

CacheFlow™ 110

Installation Guide

CacheFlow®

Configuring One Network Adapter	18
To access the Serial Console	18
To configure one network adapter using the Setup Console	19
Using the CLI to Add Network Adapters / Modify the Configuration	23
To connect to the CLI using Telnet or the Serial Console.....	23
To enable CLI for Management	24
To configure network adapters.....	24
To add DNS servers	25
To set the CacheFlow 110 name	25
To verify the CacheOS time	25
Using the Web Interface to Add Network Adapters / Modify the Configuration	26
To access the Web interface.....	26
To configure network adapters.....	28
To add DNS servers	29
To set the CacheFlow 110 name	30
To set the UTC time.....	31
Where to Go From Here	31
Security Considerations.....	32

Troubleshooting 33

Hardware Considerations	33
Software Considerations.....	33
Testing the CacheFlow 110.....	33
Cannot Access the Serial Console	33
CLI Session Times Out	33
CLI Limited to Four Sessions.....	34
Web Interface is Not Accessible	34
Web Interface Username and Password Fail.....	34
Web Interface Applets Not Displayed.....	35
Web Interface Java Errors Occur	35
User Issues.....	35
Client HTML Requests Fail	35
Client HTML Responses are Slow	37

Electrical and Environmental Specifications	39
--	-----------

CacheFlow 110 Electrical and Environment Specifications.....	39
--	----

Declaration of Conformity	43
----------------------------------	-----------

Index	45
--------------	-----------

Introduction

The CacheFlow® 110 Installation Guide provides general instructions for installing, configuring and using the CacheFlow® 110.

In particular, you can use *Troubleshooting* procedures to determine why the CacheFlow 110 does not work properly, if such is the case. In addition, you can check the CacheFlow Website for the latest support bulletins and technical notes:

<http://www.cacheflow.com>

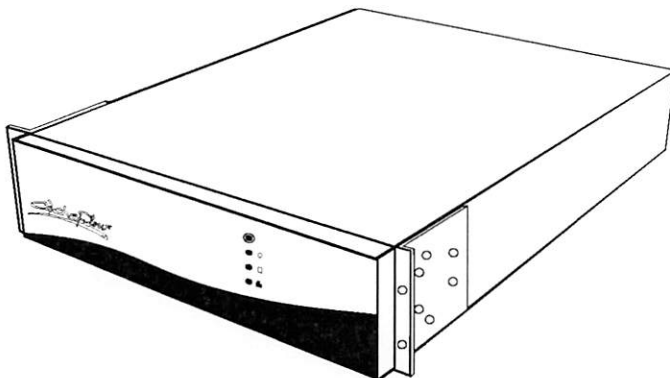
Chapter 1

Key Features

This chapter provides a general description of the CacheFlow® 110.

The CacheFlow 110 is a Web-caching appliance that accelerates Internet response time performance. The CacheFlow 110 is shipped complete, allowing for easy installation. The CacheFlow 110 can support incoming traffic loads up to 1.5 Mbps (T1) and acts as the perfect solution for small ISPs, remote POPs and for branch or remote offices.

Figure 1-1
The CacheFlow 110



The CacheFlow 110 contains these key features:

- Four gigabytes of disk storage capacity
- One power supply
- 128 megabytes of available RAM
- One network adapter

The CacheFlow 110 arrives fully assembled as a single unit. The CacheFlow 110 does not contain any user serviceable components. (Refer to the *Electrical and Environmental Specifications* at the end of this guide for specific information.)

CacheOS™ Operating System

The CacheFlow 110 runs our patent-pending, embedded CacheOS™ operating system. CacheOS represents the next generation of Web-caching technology and combines features, including Active Caching, Object Pipelining and DNS Caching, with an ultra-efficient storage system. (Refer to the *CacheOS Management and Configuration Guide* for specific information.)

Active Caching

CacheOS supports Active Caching. Active Caching describes how CacheOS stores and refreshes objects in the cache. When a Web page is requested by a client, CacheOS tracks a variety of information for every object on the page, including the frequency of requests, frequency of object modifications and time to retrieve the object. CacheOS then uses this information to determine the refresh pattern for the object.

Rather than waiting to refresh objects as clients request them, forcing clients to wait while the objects are verified, CacheOS constantly analyzes the cache and refreshes objects according to each object's refresh pattern. This drastically reduces the amount of time required to deliver objects to the clients.

Object Pipelining

CacheOS supports Pipelining. Each Web page can have dozens of objects, such as images, sounds, Java applets and so on. Under normal browser operations, the client requests a Web page, and the browser retrieves the HTML document and delivers it to the client. As the client reads the HTML document, it begins requesting the rest of the objects making up the page.

CacheFlow speeds this process. When CacheOS retrieves the HTML document, it reads the document and begins pipelining objects that make up the page. CacheOS can parse the document and request the objects much faster than the client. By the time the client begins requesting objects on the page, CacheOS has already started loading them in the cache. Object pipelining delivers complete pages to the client faster, even when the page has to be retrieved from the server.

DNS Caching

CacheOS maintains a large DNS cache by using active caching techniques similar to those already described. The DNS cache boosts overall Web cache performance. The performance improvement occurs as a result of not having to incur latencies contacting a DNS server.

Transparent Caching

CacheOS supports transparent caching. Transparent caching allows you to deploy the CacheFlow 110, without requiring users to reconfigure their browsers. This simplifies installation and ensures that users actually use the cache.

The CacheFlow 110 is configured for transparency by default. All you must do to install the CacheFlow 110 is to connect the device to your network and configure your routers or load-balancing switches to forward all packets addressed to port 80 to the CacheFlow 110.

With transparent caching, all requests that are sent to port 80 of a server are handled by the CacheFlow 110. In addition, all Web clients are supported. If you do not use transparent caching, any Web client that supports HTTP and FTP proxies can use the CacheFlow 110. Once the client defines the CacheFlow 110 as the HTTP or FTP proxy server, all requests are handled by CacheOS. You can manage multiple clients using a PAC (proxy auto-configuration) file.

Chapter 2

Installing the CacheFlow 110

This chapter describes how to unpack, assemble, install and operate the CacheFlow 110.

Unpacking the CacheFlow 110

The CacheFlow 110 arrives fully assembled, with the exception of the rack mounting brackets, which you must install.

When you receive the CacheFlow 110, verify that the following items are contained in the shipment:

Item	Quantity
CacheFlow 110 Base System, with disk drives and power supply installed	1
Rack Mounting Brackets	2
Rack Mounting Screws	8
AC Power Cord	1
Null Modem Cable	1
Serial Loopback Connector	1
<i>CacheFlow 110 Installation Guide</i>	1
<i>CacheOS Management and Configuration Guide</i>	1

If any of the items are missing or damaged, contact your CacheFlow Customer Service representative immediately.

Installing the CacheFlow 110 in an Equipment Rack

You can install the CacheFlow 110 on a shelf or on L-brackets in an equipment rack, with rack-mounting brackets securely fastened to the front of the unit.

Safety Instructions

General

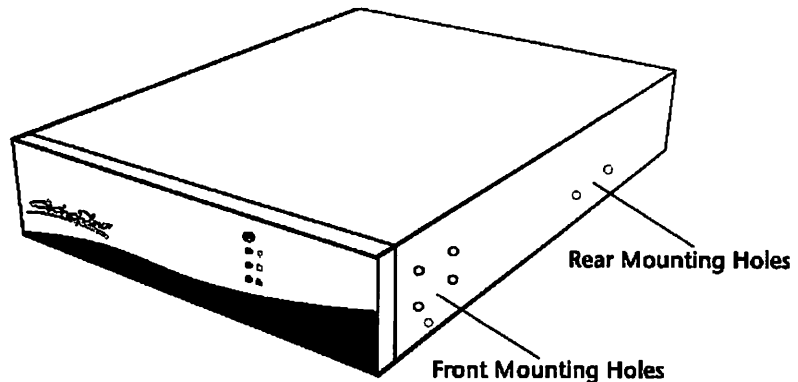
- Follow all warnings and instructions marked on the product and included in this manual.
- The CacheFlow 110 contains lithium batteries. Lithium batteries can explode if incorrectly replaced. Contact your CacheFlow Customer Service representative for assistance.

Rack Mounting



- Verify the shelf and rack are rated to support the installed equipment.
- Avoid tipping when the system tray is opened.
- Ensure that the weight is distributed properly.
- The rack must allow sufficient space to provide adequate air circulation. There should be a minimum of three inches clearance both in front of and behind the CacheFlow 110.

Figure 2-1
Rack Mounting Brackets



To install rack-mounting brackets

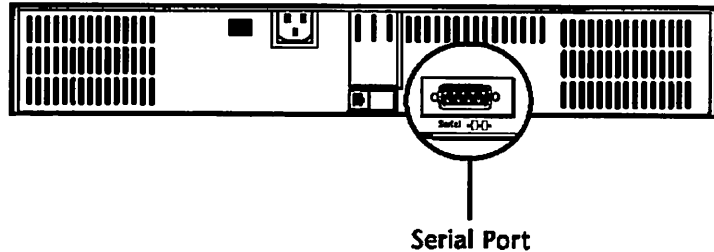
You must position each rack-mounting bracket to support the CacheFlow 110 securely in the rack and to distribute the weight properly. (Refer to Figure 2-1 to see where the front- and rear-mounting holes are.)

1. Install one bracket on each side of the CacheFlow 110 housing by using the mounting screws provided. Be sure to use all four screws for each bracket.
 - To install the CacheFlow 110 on a shelf or L-brackets in an equipment rack, secure the rack mounting brackets to the front of the unit.
 - To install the CacheFlow 110 freestanding, without a shelf or L-brackets, secure the rack mounting brackets to the middle of the unit.
2. Slide the CacheFlow 110 into place and attach the mounting brackets to the equipment rack. Be sure to use all four screws for each bracket if possible.

Connecting the CacheFlow 110 to a Terminal

The first time you start the CacheFlow 110, you must configure the network settings by using the Serial Console. You access the Serial Console through a terminal connected to the serial port of the CacheFlow 110, using the null modem cable supplied with the unit. (Refer to *Configuring One Network Adapter* in the *Configuration* chapter.)

Figure 2-2
Serial Port Connector



To connect to the Serial Console

1. Connect the null modem cable to the CacheFlow 110 serial port.
2. Connect the null modem cable to a terminal using any standard serial (RS-232) terminal. The terminal can be a directly connected terminal, a terminal connected through a terminal server or a computer running a terminal emulator.
3. Verify that the terminal is configured to the following settings:
 - Baud rate: 9600 bps
 - Data bits: 8
 - Parity: none
 - Stop bits: 1
 - Flow control: none
 - Smooth-scroll: disabled

To test the serial port configuration of the terminal

1. Connect the loop-back connector to one end of the null modem cable.
2. Connect the other end of the cable to your computer's serial port.

3. Establish a connection with your terminal emulator.
4. Type in the terminal emulator and verify that keystrokes echo on the screen. If your typing echoes on the screen, the terminal emulator is configured for the correct serial port and the serial port is functioning.
5. When the configuration is verified, remove the loop-back connector and connect the null modem cable to the CacheFlow 110 serial port.

This does not verify that the terminal emulator is configured for the correct baud rate and so on, described previously.

To resolve problems if the loop-back test fails

If you cannot type in the terminal window or if keystrokes do not echo on the screen, a problem exists with your configuration and you cannot connect to the CacheFlow 110. If the loop-back test fails, you must resolve your serial port configuration problems before attempting to connect to the CacheFlow 110.

- Verify that the COM port selected in the terminal emulator matches the COM port where the null modem cable is connected. If you are not sure which COM port is connected, try the other COM ports in the terminal emulator and see if the loop-back test succeeds.
- Verify that the COM port interrupt and base I/O address are correct and do not conflict with other devices.

Powering up the CacheFlow 110

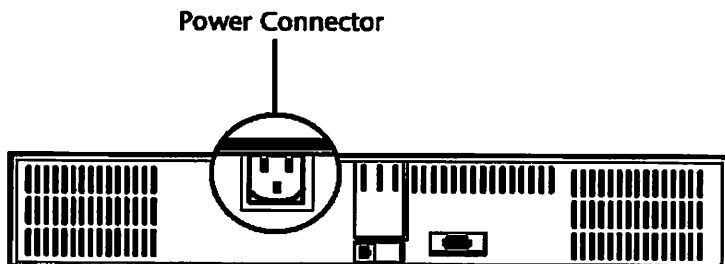


Do not attempt to maintain or tamper with the power supply.

To connect power to the unit

1. Make sure that the power switch (shown in Figure 2.4) is **not** pushed in (or in the ON position).
2. Plug the socket end of the power cord into the power connector for the power supply at the back of the CacheFlow 110, shown in Figure 2-3.
3. Plug the other end of the power cord into a grounded power source.

Figure 2-3
Power Connector



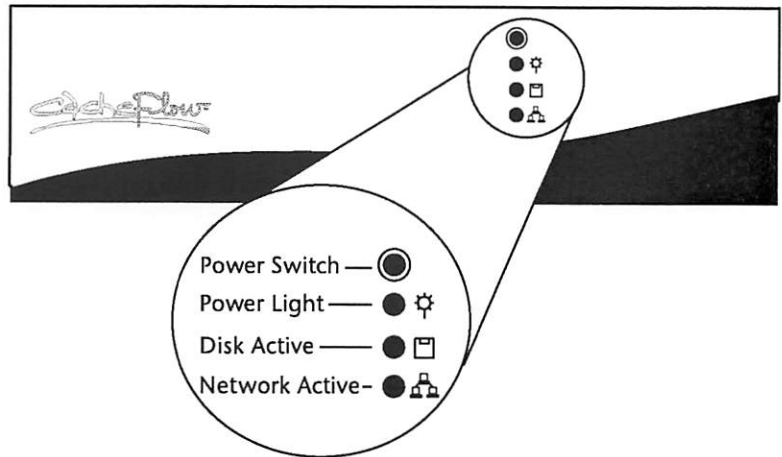
To power on the CacheFlow 110

Press the power switch, shown in Figure 2-4.

To verify that the CacheFlow 110 starts up successfully

Check the power supply and disk drive LEDs, shown in the table following Figure 2-4.

Figure 2-4
Power and Disk
Drive LEDs



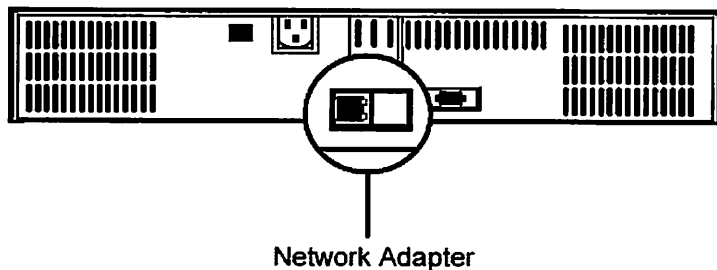
Power and Disk LEDs	Description
Power Light	The Power Light LED stays on when the associated power supply is working.
Disk Active	After the boot sequence, the Disk Activity LED flashes with disk activity.
Network Active	This LED flashes when the network connection becomes active.

Connecting the CacheFlow 110 to the Network

To connect the unit to the network

1. Connect an RJ45 Cat 5 Ethernet cable to the network adapter on one end, shown in Figure 2-5.
2. Connect the other end of the Ethernet cable to the network.

Figure 2-5
Network Adapter
Connector and LEDs



To determine the connection status of the adapter

When the network adapter is connected, you can use the adapter LEDs to determine the connection status of the adapter, shown in the following table:

Network Adapter LEDs	Description
Activity	Flashes when network activity exists. (This is the bottom, or amber, LED.)
Link	Indicates the connection status between the CacheFlow 110 and the network hub. Lights when the CacheFlow 110 is properly connected to the network. If the Link LED does not light, a problem exists with the network connection. (This is the top, or green, LED.)

Configuring the CacheFlow 110

When the CacheFlow 110 is installed and running properly, you must configure the one network adapter by using the Setup Console, accessed through the Serial Console. You must set up this adapter to connect to the Web interface and to the CLI. (Refer to *Configuring the CacheFlow 110* for more information.)

Chapter 3

Configuring the CacheFlow 110

This chapter describes how to configure the CacheFlow 110 in your environment by using CacheOS.

Using CacheOS to Configure the CacheFlow 110

Once you install the unit by using the Serial Console and Setup Console, you can use the CacheOS Web interface or command line interface (CLI) to configure the CacheFlow 110.

Serial Console

When you connect a device to a TCP/IP network, you must assign the device an IP address. When you enter the network configuration for the first time, the CacheFlow 110 uses the Setup Console to perform this initial assignment. You can access the Setup Console through the Serial Console. You can connect to the Serial Console from any serial (RS-232) terminal connection.

Command Line Interface (CLI)

The CLI allows you to manage and configure the CacheFlow 110 by using a Telnet connection or the Serial Console.

Web Interface

The Web interface is a powerful, Java-based application allowing you to manage the CacheFlow 110 from any location by using a standard Web browser. The Web interface provides an easy way to configure, monitor and upgrade the CacheOS.

Configuring One Network Adapter

Once you install the CacheFlow 110, you must configure one network adapter in order to connect to the Web interface and the CLI.

These are the required settings:

- IP Address
- Subnet Mask
- Default Gateway Address
- DNS (Domain Name Service) Server Address
- Console Username and Password

You enter “first time” settings for the network adapter by using the Setup Console, accessible from the Serial Console.

To access the Serial Console

The Serial Console allows you to access the Setup Console. To access the Serial Console, you must first connect a terminal to the CacheFlow 110 serial port. (Refer to *Connecting the CacheFlow 110 to a Terminal*.)

When you power on the CacheFlow 110, it may take from one to three minutes for the CacheFlow 110 to start, depending upon memory amount and the number of disks installed. As the device boots, information messages appear on the screen. When the boot sequence completes, the following information appears.

Note. The configuration alert appears only when the system boots the first time (before the system receives an IP address and so on), when the system is configured improperly or when the system is restored to defaults.

Example: Serial Console Starter

Starter Version 1.2

This machine has the following bootable systems:

>1: Version: 2.1.07 Release id: 12115

The default boot system is:

1: Version: 2.1.07 Release id: 12115

Press the space key to select an alternate system to boot.

Seconds remaining until the default system is booted: 0

Booting "Version: 2.1.07 Release id: 12115"

This is a system with machine id 0000FFFF06DA8942.

***** CONFIGURATION ALERT *****

One or more IP addresses have not been configured.

System startup cannot continue until the missing IP
addresses are provided.

***** SYSTEM STARTUP TEMPORARILY SUSPENDED *****

Press "enter" three times to activate the serial console

Important! When prompted, press the **Enter** key on the terminal three times to access the Serial Console. If the Serial Console does not display, verify the terminal configuration.

To configure one network adapter using the Setup Console

Use the Setup Console initially to configure any network adapters you have installed, identified by adapter location on the CacheFlow 110.

1. When the prompt appears, press Enter to use the current adapter location or enter the number of the adapter location for the network adapter you want to configure.

2. Once you enter the adapter location, assign IP addresses to the CacheOS device. When complete, each of the IP addresses you entered appears.
 - IP Address: Enter a valid IP address for your network. (This is the address you use to access the CacheFlow 110 Web interface.)
 - Subnet Mask: Enter the subnet mask for your network subnet class. (The default subnet mask is 255.255.255.0.)
 - IP Gateway Address: Enter the default IP gateway address. You use this gateway IP address for routing on the network.
 - DNS (Domain Name Service) Server. Enter the DNS Server address. You use this IP Gateway Address to resolve host names to IP addresses. You can enter additional DNS servers using the Web interface or the CLI after you have finished entering the network settings.
3. Press **Enter** to confirm the IP addresses.
4. You can enter a console user account. Enter the Console Username and Console Password, and Enable Password to prevent unauthorized access to CacheOS. The Enable Password restricts access to the enable mode of the CLI. When you enter the **enable** command in the CLI, CacheOS prompts for the Enable Password.

If you do not enter the Console Username, Console Password and Enable Password, or forget one of them, you can use the Setup Console to reset them.

5. You can restrict access to an authorized workstation. Access restrictions can limit access to the Web interface and CLI to individual workstations or subnets. From the Setup Console, you can limit access to an individual workstation only according to their IP address.

Example Setup Console Session

Notice the following example of a Setup Console session:

Welcome to the CacheOS Setup Console

----- (page 1 of 4) -----

Press <ESC> at any time to return to the main menu

DIReCTIONS:

This setup console is used to assign IP addresses to the CacheOS device. After assigning the IP addresses you can connect to the command line interface or Web interface to perform additional management tasks.

IP address [0.0.0.0]: **192.168.10.18**<Enter>
IP subnet mask [255.255.255.0]: **255.255.255.0**<Enter>
IP gateway [0.0.0.0]: **192.168.10.72**<Enter>
DNS server [0.0.0.0]: **192.168.10.64**<Enter>

You have entered the following IP addresses:

IP address: 192.168.10.18
IP subnet mask: 255.255.255.0
IP gateway: 192.168.10.72
DNS server: 192.168.10.64

Would you like to change any of them? Y/N [No] <Enter>

----- (page 2 of 4) -----

Press <ESC> at any time to return to the main menu

DIRECTIONS:

You can connect to the command line interface or Web interface to perform additional management tasks.

WARNING - access to the CacheOS device is currently unrestricted and it may be configured by unauthorized persons.

Would you like to create a console user account now? Y/N [Yes] **Y<Enter>**

Enter console username: **Admin<Enter>**

Enter console password: *******<Enter>**

Verify password: *******<Enter>**

Enter enable password: *******<Enter>**

Verify password: *******<Enter>**

----- (page 3 of 4) -----

Press <ESC> at any time to return to the main menu

DIRECTIONS:

Access to the command line interface and Web interface can be restricted to specific workstations, identified by their IP address.

This setup console allows you to add one IP address to the list of authorized workstations (additional workstations may be configured later from either the command line interface or Web interface).

The CacheOS device can currently be accessed from any workstation.

Would you like to restrict access to an authorized workstation? Y/N [No] **Y<Enter>**

Authorized workstation [0.0.0.0]: 192.168.10.56

----- (page 4 of 4) -----

DIRECTIONS:

The CacheOS device has been successfully configured to use IP address: "192.168.10.18"

You can connect to the command line interface or Web interface to perform additional management tasks. To connect to the command line interface, open the following location from your Telnet application:
192.168.10.18

To connect to the Web interface, go to the following location with your Internet browser:
<http://192.168.10.18:8081/>

The CacheOS device is password protected and may be accessed only from authorized workstations, such as 192.168.10.56

----- CONFIGURATION COMPLETE -----

Press "enter" three times to activate the serial console
 Executing image: Version: 2.1.07 Release id: 12115
 Adapter 0: MAC address 00:A0:C9:06:A8:76, 100 megabit, half duplex
 CacheFlow 110 console listening on port 8081

At this point, you can disconnect the terminal and start using the cache, or you can proceed to add other network adapters or modify the configuration, if needed.

Using the CLI to Add Network Adapters / Modify the Configuration

After you use the Setup Console to configure one network adapter, you can modify the configuration, if need be. The CLI allows you to configure one or more network adapters at the same time and modify the configuration, as needed.

When you initially configure the CacheFlow 110, perform the following tasks:

- Configure network adapters
- Add DNS servers
- Set the CacheFlow 110 name
- Verify the CacheOS time

To connect to the CLI using Telnet or the Serial Console

To connect to the CLI using Telnet, connect to the Web cache IP address (the Web cache listens for connections on the default Telnet port or port 23). After the initial configuration, the Serial Console permits you to select either the CLI or the Setup Console to configure network adapters for the CacheFlow 110 device. Refer to the following examples.

Example 1: Connecting from the Serial Console

Start the CLI by selecting 1 when the Serial Console menu is displayed:

Welcome to the CacheOS Serial Console

Version: 2.1.07 Release id: 12115

----- MENU -----

- 1) Command Line Interface
- 2) Setup Console

Enter option: 1<Enter>

Example 2: Connecting with Telnet

```
telnet> open 192.168.10.18<Enter>
username: admin<Enter>
password: *****<Enter>
192.168.10.18 - CacheFlow 110>
```

To enable CLI for Management

Once you have connected to the CLI, enter the **enable** command and then enter the **config** command:

```
192.168.10.18 - CacheFlow 110> enable
Password: *****<Enter>
192.168.10.18 - CacheFlow 110# configure<Enter>
Configuring from terminal or network [terminal]? terminal<Enter>
Enter configuration commands, one per line. End with CTRL-Z.
192.168.10.18 - CacheFlow 110#(config)
```

When you enter the **enable** command, a prompt appears, asking you for the enable password. If you have not set an enable password, just press **Enter**. When you enter the **configure** command, a prompt appears, showing you the configuration type. To configure CacheOS interactively, enter **terminal** or just press **Enter**. The **network** configuration option allows you to configure CacheOS from a configuration file located on the network.

To configure network adapters

You can configure each network adapter you have installed by using the CLI. For example, to set the IP address for adapter number one using the CLI, enter the following commands:

```
192.168.10.18 - CacheFlow 110 #(config) interface 1<Enter>
192.168.10.18 - CacheFlow 110 #(config interface 1) ip-address 192.168.10.19<Enter>
192.168.10.18 - CacheFlow 110 #(config interface 1) subnet-mask 255.255.255.0<Enter>
192.168.10.18 - CacheFlow 110 #(config interface 1) exit<Enter>
192.168.10.18 - CacheFlow 110 #(config)
```

To add DNS servers

You can define multiple DNS servers by using the CLI or Web interface. To configure DNS servers using the CLI, use the configure DNS server commands:

```
192.168.10.18 - CacheFlow 110 #(config) dns server 192.168.10.65<Enter>
```

To set the CacheFlow 110 name

You can assign a name to the CacheFlow 110 by using the configure **hostname** command. Once assigned, the name appears in the Web interface, used in the prompt for the CLI, as shown in the examples. If you have entered a host name for the CacheFlow 110 in your DNS server, use the same name.

```
192.168.10.18 - CacheFlow 110 #(config) hostname CacheFlow Demo
ok
192.168.10.18 - CacheFlow 110 #(config) exit
CacheFlow Demo#
```

To verify the CacheOS time

Verify that the NTP is correctly updating the CacheOS time by using the **show clock** command. CacheOS uses NTP to set the time; a set of default NTP servers configures it.

Show works from within the (config) menus.

```
192.168.10.18 - CacheFlow 110 # show clock<Enter>
Tue, 13 Apr 1999 03:53:41 UTC
192.168.10.18 - CacheFlow 110 #
```

If you do not set the time correctly, you can change the NTP servers by using the NTP commands in the configure mode. If you do not have access to an NTP server, you can set the time manually by using the Web interface. (See *To Set the UTC Time* in this chapter.)

Using the Web Interface to Add Network Adapters / Modify the Configuration

You can use the Web interface to add other network adapters or modify the configuration. You can connect to the Web interface by using any Web browser.

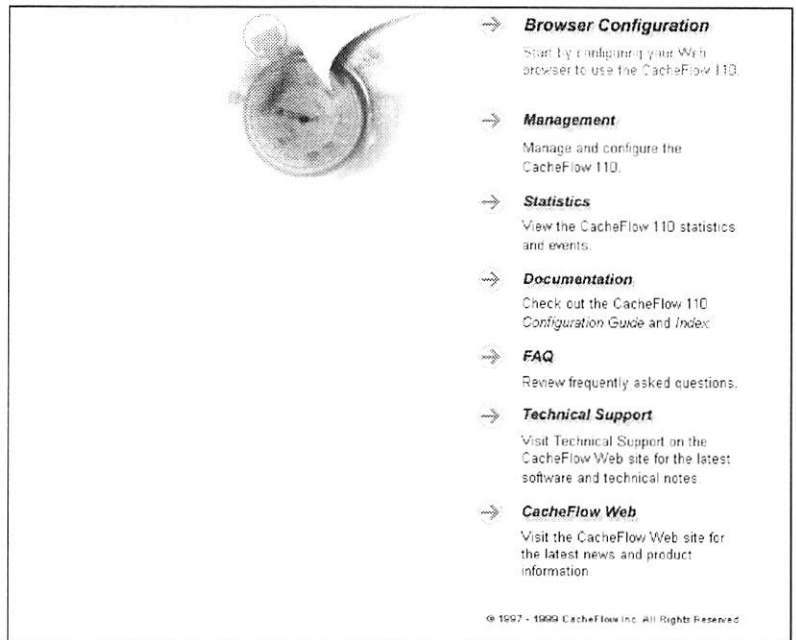
To access the Web interface

Go to the IP address of the CacheFlow 110 by using your Web browser and specify port 8081 (management port). For example, if you configured the IP address through the Serial Console with 192.168.10.18, you would enter the following URL in your Web browser:

`http://192.168.10.18:8081`

A set of Web pages and Java applets, stored on the CacheFlow 110, compose the Web interface. (CacheOS acts as a Web server on the management port to serve these pages and applets). The Web interface home page appears in Figure 3-1. From the home page, you can access the management applets, statistics applets and necessary documentation. A complete online help facility support the Web interface with a complete online help facility to assist you in defining the various configuration options. (You can find more detailed information about the Web interface in the *CacheOS Management and Configuration Guide*.)

Figure 3-1
Web Interface Home Page



When you initially configure the CacheFlow 110, perform the following tasks:

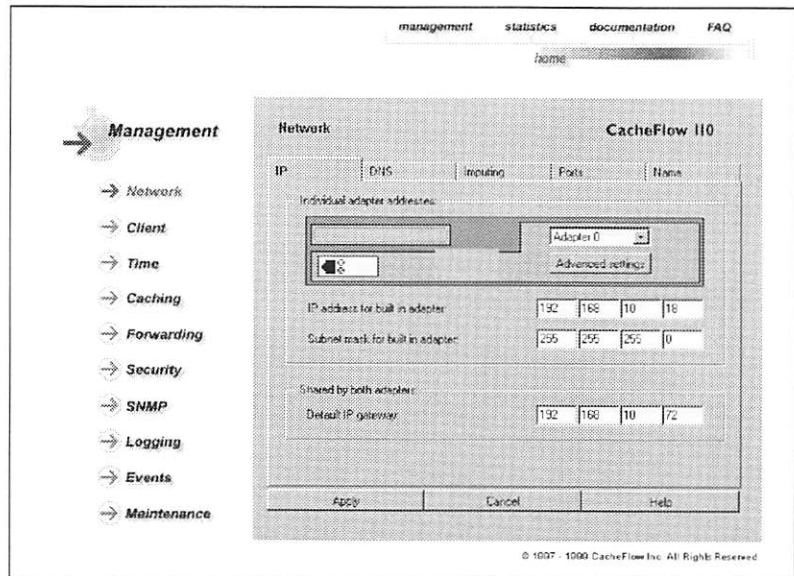
- Configure the network adapters
- Add DNS servers
- Set the CacheFlow 110 name
- Set the CacheOS time

To configure network adapters

You initially configure a network adapter by using the Setup Console. If you have multiple adapters, you must configure each one by using the CLI or Web interface.

1. Select Management from the CacheFlow 110 home page.
2. In the IP tab on the Network applet, select the network adapter to configure from the adapter drop-down list.
3. Enter the IP address and subnet mask for the adapter.
4. Click the Apply button to save your changes.

Figure 3-2
Configuring Network
Adapters

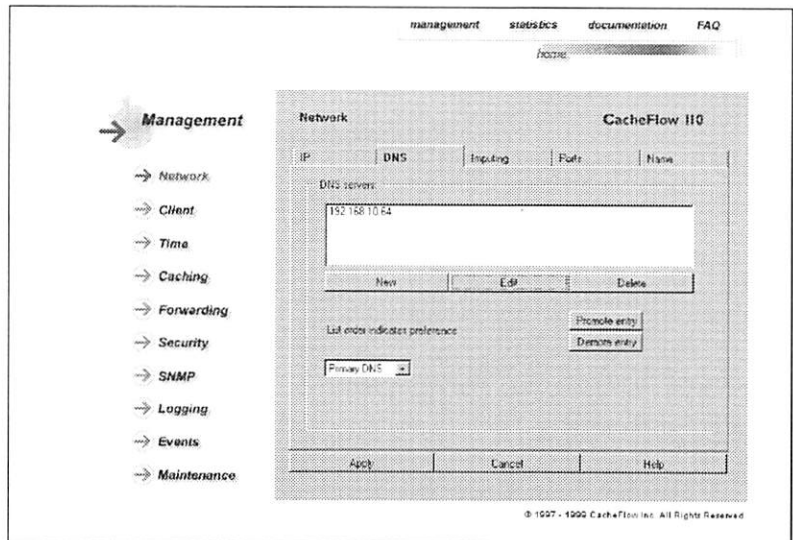


To add DNS servers

You enter the first DNS server by using the Setup Console. You can define multiple DNS servers by using the CLI or Web interface.

1. Select Management from the CacheOS home page.
2. Select the DNS tab in the Network applet.
3. To enter additional DNS servers, click the New button.
4. Type the IP address of the DNS server and click the OK button.
5. Click the Apply button to save your changes.

Figure 3-3
Adding DNS Servers

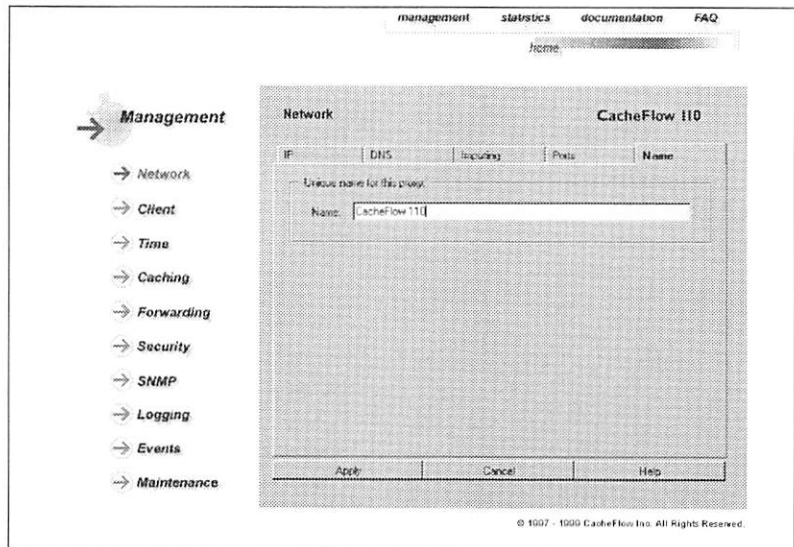


To set the CacheFlow 110 name

You can assign a name to the CacheFlow 110. The name appears in the Web interface, used for the CLI prompt. If you have entered a host name for the CacheFlow 110 in your DNS server, you should use the same name.

1. Select Management from the CacheFlow 110 home page.
2. Select the Name tab in the Network applet.
3. Enter the name in the name field.
4. Click the Apply button to save your changes.

Figure 3-4
Setting the
CacheFlow 110 Name

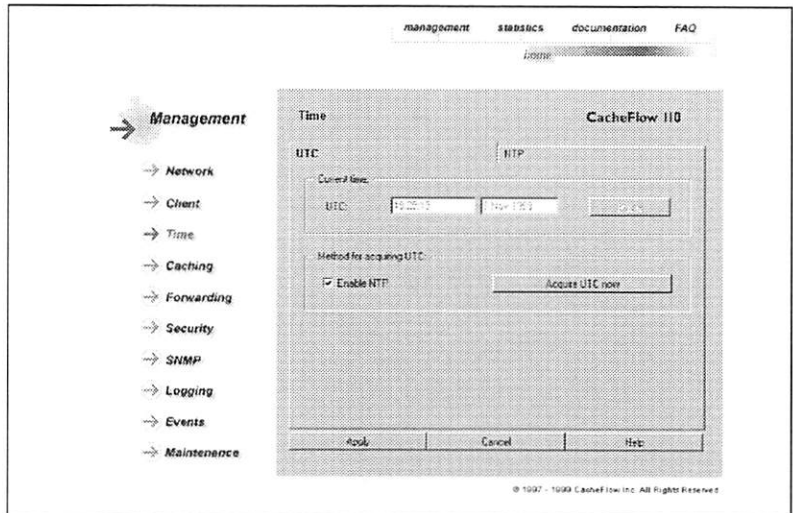


To set the UTC time

CacheOS sets UTC time by connecting to an NTP server. CacheOS includes a list of NTP servers available on the Internet. If an NTP server is not available, you can set the time manually, but you should not disable NTP, unless it is not available. To set the time manually by using the Web interface, follow these steps:

1. Select Management from the CacheFlow 110 home page.
2. Select the Time applet.
3. Clear the Enable NTP checkbox.
4. Click the Pause button.
5. Enter the current UTC time and date in the UTC fields.
6. Click the Resume button.
7. Click the Apply button to save your changes.

*Figure 3-5
Setting the Time*



Where to Go From Here

Once you install and configure the CacheFlow 110, it begins to cache Web objects. As soon as CacheFlow 110 begins running, you will see disk and network activity as it begins maintaining the cache.

Once the CacheFlow 110 becomes operational, you can configure one of the following:

- Configure your routers or load-balancing switches to forward requests transparently to the device.
- Configure the client instructions and notify users that the CacheFlow 110 is now available.

If you do not use transparent caching, you must configure each browser separately to use the CacheFlow 110. The client instructions make it easy for your users to modify their configuration. (Refer to the *CacheOS Management and Configuration Guide* for more information.)

Security Considerations

Once you configure CacheOS and it runs properly, secure the device to ensure that it is managed and used only as authorized. You can restrict access to the device using the following configuration options:

- You can limit access to the management console to specific IP addresses or subnets. To set console restrictions by using the CLI, refer to the **security** command. (To set console restrictions by using the Web interface, refer to *Setting Access Restrictions* in the *CacheOS Management and Configuration Guide*.)
- You can restrict user access to CacheOS by using the filter list ACL (access control list) options. To create an ACL using the CLI, refer to the **filter-list** command. (To create an ACL using the Web interface, refer to *Using a Filter List* in the *CacheOS Management and Configuration Guide*.)
- You can disable proxy mode to prevent unauthorized use of the cache. To turn off the proxy mode, set the proxy port to zero. To change the proxy port using the CLI, refer to the **http-proxy-port** command. (To change the proxy port using the Web interface, refer to *Setting IP Ports* in the *CacheOS Management and Configuration Guide*.)

Numerous other management options, settings and statistics exist that you can review. Refer to the *CacheOS Management and Configuration Guide* for a complete list of basic and advanced configuration procedures for using the CLI interface and Web interface.

Chapter 4

Troubleshooting

This chapter describes how to locate and solve common problems when installing and using the CacheFlow 110.

Hardware Considerations

The CacheFlow 110 arrives fully assembled as a single unit. The CacheFlow 110 does not contain any user serviceable components. If you experience problems with the proper operation of the CacheFlow 110, contact your CacheFlow Customer Service representative for assistance.

Software Considerations

Testing the CacheFlow 110

If you suspect a problem with the CacheFlow device, you can test its operation and connection using the CLI. You can use the **ping** and **traceroute** commands to test the network connection. From enable mode, you can use the **test** command to retrieve an object by using CacheOS, or run a loopback test to verify that the device is operational.

Cannot Access the Serial Console

When connecting to the Serial Console, you must configure the terminal emulator for the correct settings. You can connect to the Serial Console by using a terminal, a terminal server or a computer running a terminal emulator. For detailed instructions on connecting the terminal, go to page 10.

CLI Session Times Out

When you have a Telnet session to the CLI, that session remains open as long as activity exists. If you leave the session idle, the connection eventually times out and you have to reconnect. The default timeout is

five minutes. You can set the timeout using the **line-vty** command in configure mode.

CLI Limited to Four Sessions

Four active Telnet sessions (and one Serial Console session) limit the CLI. When you try to connect a fifth Telnet session, you will receive a message stating that no more sessions are available. To connect, you must wait for one of the active sessions to close.

Web Interface is Not Accessible

When you connect to the CacheFlow 110 Web management port from a Web browser, the Web interface should be displayed. If the Web browser fails when attempting to connect to the CacheFlow 110, follow the steps outlined below to help identify the problem:

1. Verify that you have typed the correct IP address and port (the default port is 8081) for the CacheFlow 110 in the Web browser. The only way you can verify the address the CacheFlow 110 is using without the Web interface is to connect to the Serial Console and display the network configuration. (Refer to *Connecting the CacheFlow 110 to a Terminal*, presented earlier in this guide.)
2. Verify that your workstation is configured and working properly by connecting to other Websites (such as www.cacheflow.com). If you configured your browser to use the CacheFlow 110 as the proxy server and a problem exists with the CacheFlow 110, this test may fail.
3. If you try accessing a CacheFlow 110 located on a remote network (any segment other than the segment where your workstation is attached), verify that other servers on that network are accessible.
4. Try pinging the IP address to verify that the CacheFlow 110 is accessible from the workstation. If the CacheFlow 110 does not respond to the ping, verify that it is operational, as described earlier.

Web Interface Username and Password Fail

You can protect the Web interface with a username and password. If the username and password fail, you can reset them by using the Serial Console. Go to page 18 for information on using the Serial Console.

Web Interface Applets Not Displayed

The Web interface requires Java and JavaScript. You must use a Web browser that supports Java and JavaScript, and the Web browser must have Java and JavaScript enabled. Netscape 3.0 and later versions and Internet Explorer 3.0 and later versions support Java.

If the browser encounters an error when loading an applet, restart the browser.

Web Interface Java Errors Occur

When you upgrade the Web interface, you must clear the local cache and restart the browser to guarantee that the new Java components for the Web interface are loaded. Each Web browser handles Java applets in a different way. Some browsers do not reload applets until their cache is cleared or until the browser is restarted.

User Issues

This section helps you identify common problems when the Web client either fails or is slow when retrieving Web objects.

Client HTML Requests Fail

When a request for a Web document fails, it indicates one of the following:

- A filter blocks the Web site for security or content, as when a filter blocks the source IP address or destination URL. Verify that the Website is accessible without the cache.
- The Web browser is not properly configured to use the CacheFlow 3000, as if the browser proxy settings do not point at the cache.
- The CacheFlow 3000 is not properly configured for the browser or the network. For example, the CacheFlow3000 is not configured with the correct default gateway or DNS server.
- Network problems occur, as when the Web server or the Internet connection is down.

To locate the problem, follow the steps outlined below:

1. If you use the CacheFlow 110 to access the Internet, and the CacheFlow 110 has been working properly, the most likely cause of failed requests may be the route between the

CacheFlow 110 and the Internet or intranet. Before you spend time troubleshooting the CacheFlow 110, verify that your connection to the Internet by using the ping and traceroute commands from the CLI. If you are able to ping the destination from the CacheFlow 110, use the test command to verify the device can get the object. (For information on the ping, traceroute, and test commands, refer to the *CacheOS Management and Configuration Guide*. If the request fails, try other HTML requests to verify all requests fail.)

2. The CacheFlow 110 can be configured to deny access to address groups. If the CacheFlow 110 is configured for forwarding or filtering, verify that the requested address does not match a denied subnet and mask. To configure deny settings in the Web interface, select Management from the home page and then ICP/Forwarding.
3. If your network is not configured for transparency, check the Web browser to see if it is using a PAC file for auto-configuration. If the Web browser is configured to use a PAC file, verify that the address of the PAC file is correct and that the file is accessible. (Refer to the *CacheOS Management and Configuration Guide* for additional information.) If auto-configuration is not being used, check the Web browser's proxy configuration. If you are not using transparency, you must configure the Web browser for the CacheFlow 110's IP address and port. (Refer to the *CacheOS Management and Configuration Guide* for information on configuring the clients.)
4. If the correct IP address and port for the proxy server is specified in the Web browser, try pinging the IP address to verify that the CacheFlow 110 is accessible from the workstation. If it does not respond to the ping, verify that it is operational, as described earlier. Also verify that you can ping other nodes on the network. If you can ping the CacheFlow 110, try pinging the workstation from CacheFlow 110 CLI.
5. If the CacheFlow 110 responds to the ping, verify that the CacheFlow 110's configuration is correct. If it is configured to forward requests, verify the server to which the CacheFlow 110 is forwarding requests. If the CacheFlow 110 is not forwarding requests, check the default gateway address and DNS address.
6. If the CacheFlow 110's default gateway address and DNS address are correct, try pinging each address from the CLI to verify that the servers are running. Be sure to ping the gateway

and DNS server from the same network segment where the CacheFlow 110 is connected.

7. If the default gateway is accessible, the problem most likely lies outside the local network. To verify that the problem is not associated with the CacheFlow 110, you must configure your workstation for the same gateway address as the CacheFlow 110, and configure the Web browser not to use a proxy server for HTTP requests.

Client HTML Responses are Slow

If every request from the client results in a slow response, a problem accessing a PAC file or the CacheFlow 110 can occur. To identify the problem, follow the steps outlined below:

1. Check the Web browser to see if it is using a PAC file for auto-configuration. If the Web browser is configured to use a PAC file, verify that the address of the PAC file is correct and that the file is accessible. (Refer to the *CacheOS Management and Configuration Guide* for additional information on using PAC files.) If the PAC file is not accessible, the Web browser may attempt to access the PAC file, then wait for a timeout before retrieving the object directly.
2. Check the Web browser's proxy address and port. The Web browser must be configured for the CacheFlow 110's IP address and port. (Refer to the *CacheOS Management and Configuration Guide* for information on configuring the clients.) If the address or port is not correct, the Web browser may attempt to access the CacheFlow 110, then wait for a timeout before retrieving the object directly.
3. If the correct IP address and port for the proxy server is specified in the Web browser, try pinging the IP address to verify that the CacheFlow 110 is accessible from the workstation. If the CacheFlow 110 does not respond to the ping, verify that it is operational.

Appendix A

Electrical and Environmental Specifications

CacheFlow 110 Electrical and Environment Specifications

Specifications for the CacheFlow 110 base model and a fully configured version are shown below:

	Base Model	Fully Configured
Dimensions (L x W x H)	375 x 442 x 88.1 mm 14.75 x 17.4 x 3.47 in	375 x 442 x 88.1 mm 14.75 x 17.4 x 3.47 in
Weight	9.64 kg 21.25 lb	9.64 kg 21.25 lb
Power Input	115V/230V 50 - 60 Hz	115V/230V 50 - 60 Hz
Power Consumption	100 W	100 W
Operating Temperature	41 - 122° F 5 - 50° C	41 - 122° F 5 - 50° C
Operating Humidity	< 90 % relative humidity, non-condensing	< 90 % relative humidity, non-condensing
Operating Altitude	Up to 8000 ft	Up to 8000 ft
EMC	FCC Class B EN 55022 A EN 50082-1 Iec 801-2	FCC Class B EN 55022 A EN 50082-1 Iec 801-2

	Base Model	Fully Configured
	lec 801-3 lec 801-4	lec 801-3 lec 801-4
Safety	CSA C22.2 No. 950 M95 UL 1950 3 Edition lec 950	CSA C22.2 No. 950 M95 UL 1950 3 Edition lec 950

Regulatory Statements

General

IMPORTANT: Any modifications to the unit, unless expressly approved by CacheFlow, could void the your authority to operate the equipment.

Battery Warning

CAUTION: Danger of explosion if battery is incorrectly replaced. Replace the battery only with the same or equivalent type recommended by the manufacturer. Dispose of used batteries according to the manufacturer's instructions.

Class B Digital Warning

This equipment has been tested and found to comply with the limits for a Class B digital device, pursuant to Part 15 of the FCC rules. These limits are designed to provide reasonable protection against harmful interference when the equipment is operated in a commercial environment. This equipment generates, uses and can radiate radio frequency energy, and if not installed and used in accordance with the instruction manual, may cause harmful interference to radio communications. Operation of this equipment in a residential area is likely to cause harmful interference, in which case you will be required to correct the interference at your own expense.

EMC Warning

This is a Class B product. In a domestic environment this product may cause radio interference in which case the user may be required to take adequate measures.

Declaration of Conformity



CacheFlow Inc.
650 Almanor Avenue
Sunnyvale, CA 94086

408.225.2200
408.225.2250 Fax

www.cacheflow.com

Declaration of Conformity

We, CacheFlow Inc.
650 Almanor Avenue
Sunnyvale, CA 94086 USA

Declare under our sole responsibility that the products

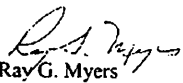
Cacheflow Model 110

To which this declaration relates is in conformity with the following
standards:

EN 60950: 1992
EN55022: 1994
EN50082-1: 1992
IEC 801-2
IEC 801-3
IEC 801-4

Following the provisions of the 73/23/EEC and 89/336/EEC Directives,
Including the Amending Directive 93/68/EEC

Sunnyvale, CA 94086 USA
Date 10-22-99


Ray G. Myers
Vice President, Manufacturing
And Customer Support

Index

A

Active Caching, 4

B

baud rate, 10

C

CacheFlow 110

serial port, 10

weight, 8

CacheFlow website, 1

CacheOS, 4

client, 5

HTML requests fail, 35

HTML responses slow, 37

Console password

setting with the Serial

Console, 20

D

data bits, 10

Declaration of Conformity, 43

default gateway, 18

default gateway address

setting with the Serial

Console, 20

dimensions, 39

DNS, 29

DNS cache, 5

DNS server, 18

setting with the Serial

Console, 20

E

equipment rack, 8

rack mounting brackets, 9

warning, 8

Ethernet adapters, 24, 25, 28

Ethernet cable, 14

F

flow control, 10

H

hardware, 3

HTML requests fail, 35

HTML responses slow, 37

I

installation

connecting a terminal, 10

connecting power, 12

connecting the network, 14

mounting in an equipment

rack, 8

network settings, 18

IP address, 18, 24, 25, 28

setting with the Serial

Console, 20

L

lithium batteries, 8

M

Management Console, 17, 26
 accessing, 23
 Java applets, 35
 not accessible, 34
 password fails, 35

N

name, 25, 30
network adapters, 24, 25, 28
network connection, 14
network settings, 18
NTP, 31

O

object pipelining, 4
On/Off switch, 12

P

packing list, 7
parity, 10
pipelining, 4
power consumption, 39
power cord, 12
power supply LED, 13
power switch, 13

R

rack mounting brackets, 9

S

safety instructions, 8
Serial Console, 17, 18
 accessing, 18
 cannot access, 33
 connecting a terminal, 10
 sample session, 21
serial port, 10
setting the time, 31
specifications, 39
stop bits, 10
subnet mask, 18, 24, 25, 28
 setting with the Serial
 Console, 20

T

terminal, 10
 settings, 10
terminal emulator, 10
terminal server, 10
time, 31

U

unpacking, 7

CacheFlow™ 110

Installation Guide

CacheFlow®

Configuring One Network Adapter	18
To access the Serial Console	18
To configure one network adapter using the Setup Console	19
Using the CLI to Add Network Adapters / Modify the Configuration	23
To connect to the CLI using Telnet or the Serial Console.....	23
To enable CLI for Management	24
To configure network adapters.....	24
To add DNS servers	25
To set the CacheFlow 110 name	25
To verify the CacheOS time	25
Using the Web Interface to Add Network Adapters / Modify the Configuration	26
To access the Web interface.....	26
To configure network adapters.....	28
To add DNS servers	29
To set the CacheFlow 110 name	30
To set the UTC time.....	31
Where to Go From Here	31
Security Considerations.....	32

Troubleshooting 33

Hardware Considerations	33
Software Considerations.....	33
Testing the CacheFlow 110.....	33
Cannot Access the Serial Console.....	33
CLI Session Times Out	33
CLI Limited to Four Sessions.....	34
Web Interface is Not Accessible	34
Web Interface Username and Password Fail.....	34
Web Interface Applets Not Displayed.....	35
Web Interface Java Errors Occur	35
User Issues.....	35
Client HTML Requests Fail	35
Client HTML Responses are Slow	37

Electrical and Environmental Specifications	39
CacheFlow 110 Electrical and Environment Specifications.....	39
Declaration of Conformity	43
Index	45

Introduction

The CacheFlow® 110 Installation Guide provides general instructions for installing, configuring and using the CacheFlow® 110.

In particular, you can use *Troubleshooting* procedures to determine why the CacheFlow 110 does not work properly, if such is the case. In addition, you can check the CacheFlow Website for the latest support bulletins and technical notes:

<http://www.cacheflow.com>

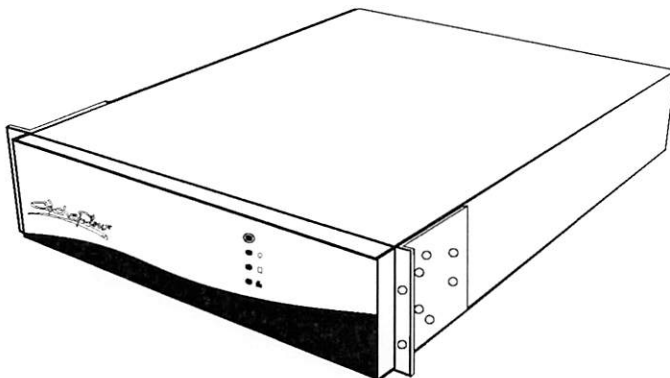
Chapter 1

Key Features

This chapter provides a general description of the CacheFlow® 110.

The CacheFlow 110 is a Web-caching appliance that accelerates Internet response time performance. The CacheFlow 110 is shipped complete, allowing for easy installation. The CacheFlow 110 can support incoming traffic loads up to 1.5 Mbps (T1) and acts as the perfect solution for small ISPs, remote POPs and for branch or remote offices.

Figure 1-1
The CacheFlow 110



The CacheFlow 110 contains these key features:

- Four gigabytes of disk storage capacity
- One power supply
- 128 megabytes of available RAM
- One network adapter

The CacheFlow 110 arrives fully assembled as a single unit. The CacheFlow 110 does not contain any user serviceable components. (Refer to the *Electrical and Environmental Specifications* at the end of this guide for specific information.)

CacheOS™ Operating System

The CacheFlow 110 runs our patent-pending, embedded CacheOS™ operating system. CacheOS represents the next generation of Web-caching technology and combines features, including Active Caching, Object Pipelining and DNS Caching, with an ultra-efficient storage system. (Refer to the *CacheOS Management and Configuration Guide* for specific information.)

Active Caching

CacheOS supports Active Caching. Active Caching describes how CacheOS stores and refreshes objects in the cache. When a Web page is requested by a client, CacheOS tracks a variety of information for every object on the page, including the frequency of requests, frequency of object modifications and time to retrieve the object. CacheOS then uses this information to determine the refresh pattern for the object.

Rather than waiting to refresh objects as clients request them, forcing clients to wait while the objects are verified, CacheOS constantly analyzes the cache and refreshes objects according to each object's refresh pattern. This drastically reduces the amount of time required to deliver objects to the clients.

Object Pipelining

CacheOS supports Pipelining. Each Web page can have dozens of objects, such as images, sounds, Java applets and so on. Under normal browser operations, the client requests a Web page, and the browser retrieves the HTML document and delivers it to the client. As the client reads the HTML document, it begins requesting the rest of the objects making up the page.

CacheFlow speeds this process. When CacheOS retrieves the HTML document, it reads the document and begins pipelining objects that make up the page. CacheOS can parse the document and request the objects much faster than the client. By the time the client begins requesting objects on the page, CacheOS has already started loading them in the cache. Object pipelining delivers complete pages to the client faster, even when the page has to be retrieved from the server.

DNS Caching

CacheOS maintains a large DNS cache by using active caching techniques similar to those already described. The DNS cache boosts overall Web cache performance. The performance improvement occurs as a result of not having to incur latencies contacting a DNS server.

Transparent Caching

CacheOS supports transparent caching. Transparent caching allows you to deploy the CacheFlow 110, without requiring users to reconfigure their browsers. This simplifies installation and ensures that users actually use the cache.

The CacheFlow 110 is configured for transparency by default. All you must do to install the CacheFlow 110 is to connect the device to your network and configure your routers or load-balancing switches to forward all packets addressed to port 80 to the CacheFlow 110.

With transparent caching, all requests that are sent to port 80 of a server are handled by the CacheFlow 110. In addition, all Web clients are supported. If you do not use transparent caching, any Web client that supports HTTP and FTP proxies can use the CacheFlow 110. Once the client defines the CacheFlow 110 as the HTTP or FTP proxy server, all requests are handled by CacheOS. You can manage multiple clients using a PAC (proxy auto-configuration) file.

Chapter 2

Installing the CacheFlow 110

This chapter describes how to unpack, assemble, install and operate the CacheFlow 110.

Unpacking the CacheFlow 110

The CacheFlow 110 arrives fully assembled, with the exception of the rack mounting brackets, which you must install.

When you receive the CacheFlow 110, verify that the following items are contained in the shipment:

Item	Quantity
CacheFlow 110 Base System, with disk drives and power supply installed	1
Rack Mounting Brackets	2
Rack Mounting Screws	8
AC Power Cord	1
Null Modem Cable	1
Serial Loopback Connector	1
<i>CacheFlow 110 Installation Guide</i>	1
<i>CacheOS Management and Configuration Guide</i>	1

If any of the items are missing or damaged, contact your CacheFlow Customer Service representative immediately.

Installing the CacheFlow 110 in an Equipment Rack

You can install the CacheFlow 110 on a shelf or on L-brackets in an equipment rack, with rack-mounting brackets securely fastened to the front of the unit.

Safety Instructions

General

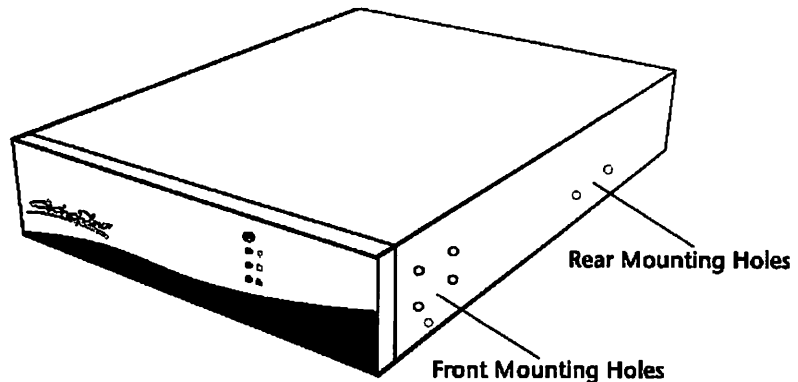
- Follow all warnings and instructions marked on the product and included in this manual.
- The CacheFlow 110 contains lithium batteries. Lithium batteries can explode if incorrectly replaced. Contact your CacheFlow Customer Service representative for assistance.

Rack Mounting



- Verify the shelf and rack are rated to support the installed equipment.
- Avoid tipping when the system tray is opened.
- Ensure that the weight is distributed properly.
- The rack must allow sufficient space to provide adequate air circulation. There should be a minimum of three inches clearance both in front of and behind the CacheFlow 110.

Figure 2-1
Rack Mounting Brackets



To install rack-mounting brackets

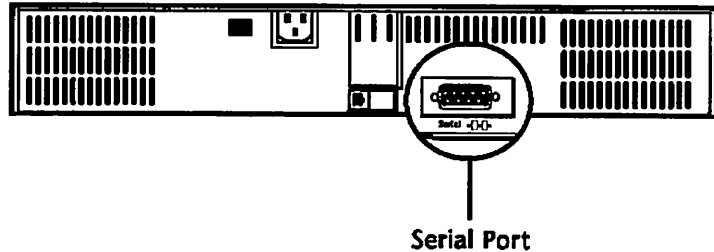
You must position each rack-mounting bracket to support the CacheFlow 110 securely in the rack and to distribute the weight properly. (Refer to Figure 2-1 to see where the front- and rear-mounting holes are.)

1. Install one bracket on each side of the CacheFlow 110 housing by using the mounting screws provided. Be sure to use all four screws for each bracket.
 - To install the CacheFlow 110 on a shelf or L-brackets in an equipment rack, secure the rack mounting brackets to the front of the unit.
 - To install the CacheFlow 110 freestanding, without a shelf or L-brackets, secure the rack mounting brackets to the middle of the unit.
2. Slide the CacheFlow 110 into place and attach the mounting brackets to the equipment rack. Be sure to use all four screws for each bracket if possible.

Connecting the CacheFlow 110 to a Terminal

The first time you start the CacheFlow 110, you must configure the network settings by using the Serial Console. You access the Serial Console through a terminal connected to the serial port of the CacheFlow 110, using the null modem cable supplied with the unit. (Refer to *Configuring One Network Adapter* in the *Configuration* chapter.)

Figure 2-2
Serial Port Connector



To connect to the Serial Console

1. Connect the null modem cable to the CacheFlow 110 serial port.
2. Connect the null modem cable to a terminal using any standard serial (RS-232) terminal. The terminal can be a directly connected terminal, a terminal connected through a terminal server or a computer running a terminal emulator.
3. Verify that the terminal is configured to the following settings:
 - Baud rate: 9600 bps
 - Data bits: 8
 - Parity: none
 - Stop bits: 1
 - Flow control: none
 - Smooth-scroll: disabled

To test the serial port configuration of the terminal

1. Connect the loop-back connector to one end of the null modem cable.
2. Connect the other end of the cable to your computer's serial port.

3. Establish a connection with your terminal emulator.
4. Type in the terminal emulator and verify that keystrokes echo on the screen. If your typing echoes on the screen, the terminal emulator is configured for the correct serial port and the serial port is functioning.
5. When the configuration is verified, remove the loop-back connector and connect the null modem cable to the CacheFlow 110 serial port.

This does not verify that the terminal emulator is configured for the correct baud rate and so on, described previously.

To resolve problems if the loop-back test fails

If you cannot type in the terminal window or if keystrokes do not echo on the screen, a problem exists with your configuration and you cannot connect to the CacheFlow 110. If the loop-back test fails, you must resolve your serial port configuration problems before attempting to connect to the CacheFlow 110.

- Verify that the COM port selected in the terminal emulator matches the COM port where the null modem cable is connected. If you are not sure which COM port is connected, try the other COM ports in the terminal emulator and see if the loop-back test succeeds.
- Verify that the COM port interrupt and base I/O address are correct and do not conflict with other devices.

Powering up the CacheFlow 110

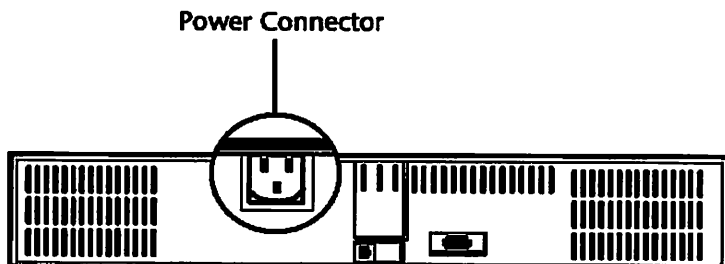


Do not attempt to maintain or tamper with the power supply.

To connect power to the unit

1. Make sure that the power switch (shown in Figure 2.4) is **not** pushed in (or in the ON position).
2. Plug the socket end of the power cord into the power connector for the power supply at the back of the CacheFlow 110, shown in Figure 2-3.
3. Plug the other end of the power cord into a grounded power source.

Figure 2-3
Power Connector



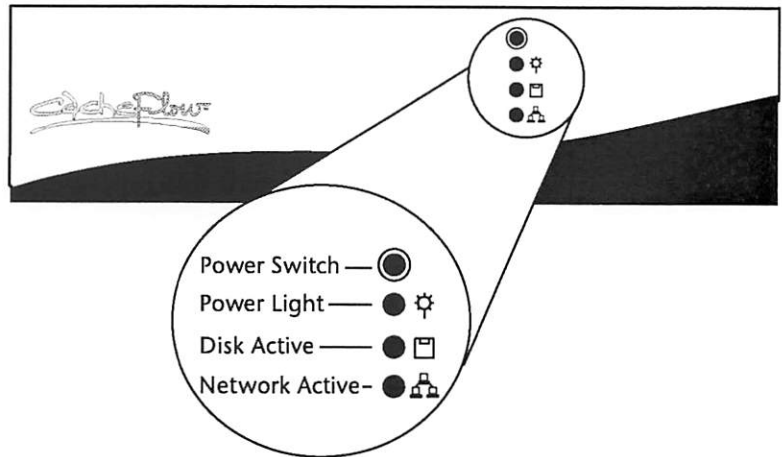
To power on the CacheFlow 110

Press the power switch, shown in Figure 2-4.

To verify that the CacheFlow 110 starts up successfully

Check the power supply and disk drive LEDs, shown in the table following Figure 2-4.

Figure 2-4
Power and Disk
Drive LEDs



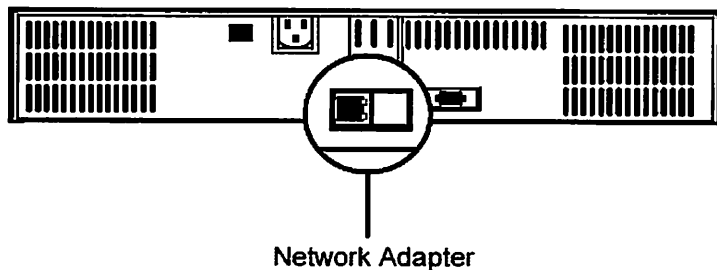
Power and Disk LEDs	Description
Power Light	The Power Light LED stays on when the associated power supply is working.
Disk Active	After the boot sequence, the Disk Activity LED flashes with disk activity.
Network Active	This LED flashes when the network connection becomes active.

Connecting the CacheFlow 110 to the Network

To connect the unit to the network

1. Connect an RJ45 Cat 5 Ethernet cable to the network adapter on one end, shown in Figure 2-5.
2. Connect the other end of the Ethernet cable to the network.

Figure 2-5
Network Adapter
Connector and LEDs



To determine the connection status of the adapter

When the network adapter is connected, you can use the adapter LEDs to determine the connection status of the adapter, shown in the following table:

Network Adapter LEDs	Description
Activity	Flashes when network activity exists. (This is the bottom, or amber, LED.)
Link	Indicates the connection status between the CacheFlow 110 and the network hub. Lights when the CacheFlow 110 is properly connected to the network. If the Link LED does not light, a problem exists with the network connection. (This is the top, or green, LED.)

Configuring the CacheFlow 110

When the CacheFlow 110 is installed and running properly, you must configure the one network adapter by using the Setup Console, accessed through the Serial Console. You must set up this adapter to connect to the Web interface and to the CLI. (Refer to *Configuring the CacheFlow 110* for more information.)

Chapter 3

Configuring the CacheFlow 110

This chapter describes how to configure the CacheFlow 110 in your environment by using CacheOS.

Using CacheOS to Configure the CacheFlow 110

Once you install the unit by using the Serial Console and Setup Console, you can use the CacheOS Web interface or command line interface (CLI) to configure the CacheFlow 110.

Serial Console

When you connect a device to a TCP/IP network, you must assign the device an IP address. When you enter the network configuration for the first time, the CacheFlow 110 uses the Setup Console to perform this initial assignment. You can access the Setup Console through the Serial Console. You can connect to the Serial Console from any serial (RS-232) terminal connection.

Command Line Interface (CLI)

The CLI allows you to manage and configure the CacheFlow 110 by using a Telnet connection or the Serial Console.

Web Interface

The Web interface is a powerful, Java-based application allowing you to manage the CacheFlow 110 from any location by using a standard Web browser. The Web interface provides an easy way to configure, monitor and upgrade the CacheOS.

Configuring One Network Adapter

Once you install the CacheFlow 110, you must configure one network adapter in order to connect to the Web interface and the CLI.

These are the required settings:

- IP Address
- Subnet Mask
- Default Gateway Address
- DNS (Domain Name Service) Server Address
- Console Username and Password

You enter “first time” settings for the network adapter by using the Setup Console, accessible from the Serial Console.

To access the Serial Console

The Serial Console allows you to access the Setup Console. To access the Serial Console, you must first connect a terminal to the CacheFlow 110 serial port. (Refer to *Connecting the CacheFlow 110 to a Terminal*.)

When you power on the CacheFlow 110, it may take from one to three minutes for the CacheFlow 110 to start, depending upon memory amount and the number of disks installed. As the device boots, information messages appear on the screen. When the boot sequence completes, the following information appears.

Note. The configuration alert appears only when the system boots the first time (before the system receives an IP address and so on), when the system is configured improperly or when the system is restored to defaults.

Example: Serial Console Starter

Starter Version 1.2

This machine has the following bootable systems:

>1: Version: 2.1.07 Release id: 12115

The default boot system is:

1: Version: 2.1.07 Release id: 12115

Press the space key to select an alternate system to boot.

Seconds remaining until the default system is booted: 0

Booting "Version: 2.1.07 Release id: 12115"

This is a system with machine id 0000FFFF06DA8942.

***** CONFIGURATION ALERT *****

One or more IP addresses have not been configured.
System startup cannot continue until the missing IP
addresses are provided.

***** SYSTEM STARTUP TEMPORARILY SUSPENDED *****

Press "enter" three times to activate the serial console

Important! When prompted, press the **Enter** key on the terminal three times to access the Serial Console. If the Serial Console does not display, verify the terminal configuration.

To configure one network adapter using the Setup Console

Use the Setup Console initially to configure any network adapters you have installed, identified by adapter location on the CacheFlow 110.

1. When the prompt appears, press Enter to use the current adapter location or enter the number of the adapter location for the network adapter you want to configure.

2. Once you enter the adapter location, assign IP addresses to the CacheOS device. When complete, each of the IP addresses you entered appears.
 - IP Address: Enter a valid IP address for your network. (This is the address you use to access the CacheFlow 110 Web interface.)
 - Subnet Mask: Enter the subnet mask for your network subnet class. (The default subnet mask is 255.255.255.0.)
 - IP Gateway Address: Enter the default IP gateway address. You use this gateway IP address for routing on the network.
 - DNS (Domain Name Service) Server. Enter the DNS Server address. You use this IP Gateway Address to resolve host names to IP addresses. You can enter additional DNS servers using the Web interface or the CLI after you have finished entering the network settings.
3. Press **Enter** to confirm the IP addresses.
4. You can enter a console user account. Enter the Console Username and Console Password, and Enable Password to prevent unauthorized access to CacheOS. The Enable Password restricts access to the enable mode of the CLI. When you enter the **enable** command in the CLI, CacheOS prompts for the Enable Password.

If you do not enter the Console Username, Console Password and Enable Password, or forget one of them, you can use the Setup Console to reset them.

5. You can restrict access to an authorized workstation. Access restrictions can limit access to the Web interface and CLI to individual workstations or subnets. From the Setup Console, you can limit access to an individual workstation only according to their IP address.

Example Setup Console Session

Notice the following example of a Setup Console session:

Welcome to the CacheOS Setup Console

----- (page 1 of 4) -----

Press <ESC> at any time to return to the main menu

DIReCTIONS:

This setup console is used to assign IP addresses to the CacheOS device. After assigning the IP addresses you can connect to the command line interface or Web interface to perform additional management tasks.

IP address [0.0.0.0]: **192.168.10.18**<Enter>
IP subnet mask [255.255.255.0]: **255.255.255.0**<Enter>
IP gateway [0.0.0.0]: **192.168.10.72**<Enter>
DNS server [0.0.0.0]: **192.168.10.64**<Enter>

You have entered the following IP addresses:

IP address: 192.168.10.18
IP subnet mask: 255.255.255.0
IP gateway: 192.168.10.72
DNS server: 192.168.10.64

Would you like to change any of them? Y/N [No] <Enter>

----- (page 2 of 4) -----

Press <ESC> at any time to return to the main menu

DIRECTIONS:

You can connect to the command line interface or Web interface to perform additional management tasks.

WARNING - access to the CacheOS device is currently unrestricted and it may be configured by unauthorized persons.

Would you like to create a console user account now? Y/N [Yes] **Y<Enter>**

Enter console username: **Admin<Enter>**

Enter console password: *******<Enter>**

Verify password: *******<Enter>**

Enter enable password: *******<Enter>**

Verify password: *******<Enter>**

----- (page 3 of 4) -----

Press <ESC> at any time to return to the main menu

DIRECTIONS:

Access to the command line interface and Web interface can be restricted to specific workstations, identified by their IP address.

This setup console allows you to add one IP address to the list of authorized workstations (additional workstations may be configured later from either the command line interface or Web interface).

The CacheOS device can currently be accessed from any workstation.

Would you like to restrict access to an authorized workstation? Y/N [No] **Y<Enter>**

Authorized workstation [0.0.0.0]: 192.168.10.56

----- (page 4 of 4) -----

DIRECTIONS:

The CacheOS device has been successfully configured to use IP address: "192.168.10.18"

You can connect to the command line interface or Web interface to perform additional management tasks. To connect to the command line interface, open the following location from your Telnet application:
192.168.10.18

To connect to the Web interface, go to the following location with your Internet browser:
<http://192.168.10.18:8081/>

The CacheOS device is password protected and may be accessed only from authorized workstations, such as 192.168.10.56

----- CONFIGURATION COMPLETE -----

Press "enter" three times to activate the serial console
 Executing image: Version: 2.1.07 Release id: 12115
 Adapter 0: MAC address 00:A0:C9:06:A8:76, 100 megabit, half duplex
 CacheFlow 110 console listening on port 8081

At this point, you can disconnect the terminal and start using the cache, or you can proceed to add other network adapters or modify the configuration, if needed.

Using the CLI to Add Network Adapters / Modify the Configuration

After you use the Setup Console to configure one network adapter, you can modify the configuration, if need be. The CLI allows you to configure one or more network adapters at the same time and modify the configuration, as needed.

When you initially configure the CacheFlow 110, perform the following tasks:

- Configure network adapters
- Add DNS servers
- Set the CacheFlow 110 name
- Verify the CacheOS time

To connect to the CLI using Telnet or the Serial Console

To connect to the CLI using Telnet, connect to the Web cache IP address (the Web cache listens for connections on the default Telnet port or port 23). After the initial configuration, the Serial Console permits you to select either the CLI or the Setup Console to configure network adapters for the CacheFlow 110 device. Refer to the following examples.

Example 1: Connecting from the Serial Console

Start the CLI by selecting 1 when the Serial Console menu is displayed:

Welcome to the CacheOS Serial Console

Version: 2.1.07 Release id: 12115

----- MENU -----

- 1) Command Line Interface
- 2) Setup Console

Enter option: 1<Enter>

Example 2: Connecting with Telnet

```
telnet> open 192.168.10.18<Enter>
username: admin<Enter>
password: *****<Enter>
192.168.10.18 - CacheFlow 110>
```

To enable CLI for Management

Once you have connected to the CLI, enter the **enable** command and then enter the **config** command:

```
192.168.10.18 - CacheFlow 110> enable
Password: *****<Enter>
192.168.10.18 - CacheFlow 110# configure<Enter>
Configuring from terminal or network [terminal]? terminal<Enter>
Enter configuration commands, one per line. End with CTRL-Z.
192.168.10.18 - CacheFlow 110#(config)
```

When you enter the **enable** command, a prompt appears, asking you for the enable password. If you have not set an enable password, just press **Enter**. When you enter the **configure** command, a prompt appears, showing you the configuration type. To configure CacheOS interactively, enter **terminal** or just press **Enter**. The **network** configuration option allows you to configure CacheOS from a configuration file located on the network.

To configure network adapters

You can configure each network adapter you have installed by using the CLI. For example, to set the IP address for adapter number one using the CLI, enter the following commands:

```

192.168.10.18 - CacheFlow 110 #(config) interface 1<Enter>
192.168.10.18 - CacheFlow 110 #(config interface 1) ip-address 192.168.10.19<Enter>
192.168.10.18 - CacheFlow 110 #(config interface 1) subnet-mask 255.255.255.0<Enter>
192.168.10.18 - CacheFlow 110 #(config interface 1) exit<Enter>
192.168.10.18 - CacheFlow 110 #(config)

```

To add DNS servers

You can define multiple DNS servers by using the CLI or Web interface. To configure DNS servers using the CLI, use the configure DNS server commands:

```

192.168.10.18 - CacheFlow 110 #(config) dns server 192.168.10.65<Enter>

```

To set the CacheFlow 110 name

You can assign a name to the CacheFlow 110 by using the configure **hostname** command. Once assigned, the name appears in the Web interface, used in the prompt for the CLI, as shown in the examples. If you have entered a host name for the CacheFlow 110 in your DNS server, use the same name.

```

192.168.10.18 - CacheFlow 110 #(config) hostname CacheFlow Demo
ok
192.168.10.18 - CacheFlow 110 #(config) exit
CacheFlow Demo#

```

To verify the CacheOS time

Verify that the NTP is correctly updating the CacheOS time by using the **show clock** command. CacheOS uses NTP to set the time; a set of default NTP servers configures it.

Show works from within the (config) menus.

```

192.168.10.18 - CacheFlow 110 # show clock<Enter>
Tue, 13 Apr 1999 03:53:41 UTC
192.168.10.18 - CacheFlow 110 #

```

If you do not set the time correctly, you can change the NTP servers by using the NTP commands in the configure mode. If you do not have access to an NTP server, you can set the time manually by using the Web interface. (See *To Set the UTC Time* in this chapter.)

Using the Web Interface to Add Network Adapters / Modify the Configuration

You can use the Web interface to add other network adapters or modify the configuration. You can connect to the Web interface by using any Web browser.

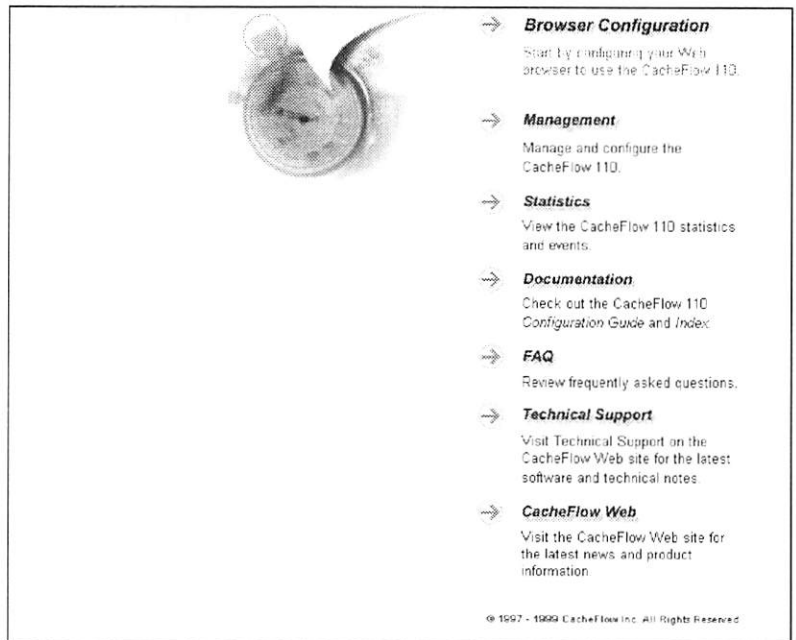
To access the Web interface

Go to the IP address of the CacheFlow 110 by using your Web browser and specify port 8081 (management port). For example, if you configured the IP address through the Serial Console with 192.168.10.18, you would enter the following URL in your Web browser:

`http://192.168.10.18:8081`

A set of Web pages and Java applets, stored on the CacheFlow 110, compose the Web interface. (CacheOS acts as a Web server on the management port to serve these pages and applets). The Web interface home page appears in Figure 3-1. From the home page, you can access the management applets, statistics applets and necessary documentation. A complete online help facility support the Web interface with a complete online help facility to assist you in defining the various configuration options. (You can find more detailed information about the Web interface in the *CacheOS Management and Configuration Guide*.)

Figure 3-1
Web Interface Home Page



When you initially configure the CacheFlow 110, perform the following tasks:

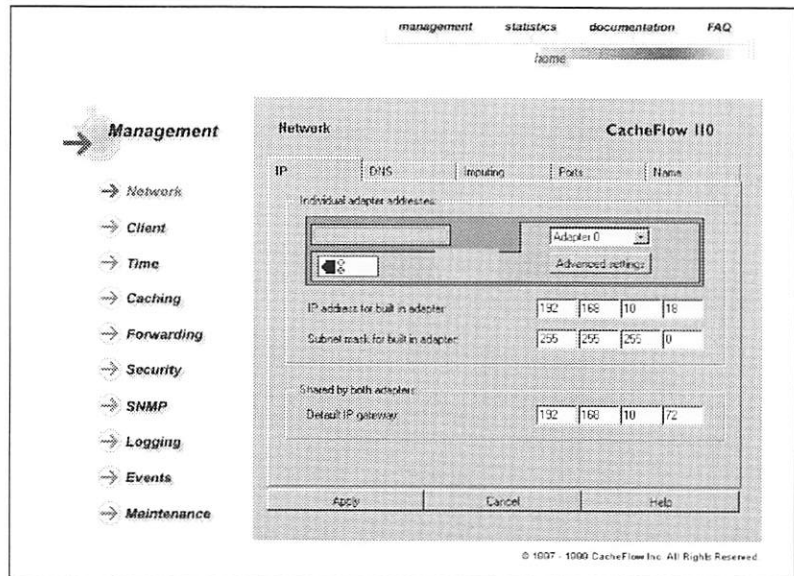
- Configure the network adapters
- Add DNS servers
- Set the CacheFlow 110 name
- Set the CacheOS time

To configure network adapters

You initially configure a network adapter by using the Setup Console. If you have multiple adapters, you must configure each one by using the CLI or Web interface.

1. Select Management from the CacheFlow 110 home page.
2. In the IP tab on the Network applet, select the network adapter to configure from the adapter drop-down list.
3. Enter the IP address and subnet mask for the adapter.
4. Click the Apply button to save your changes.

Figure 3-2
Configuring Network
Adapters

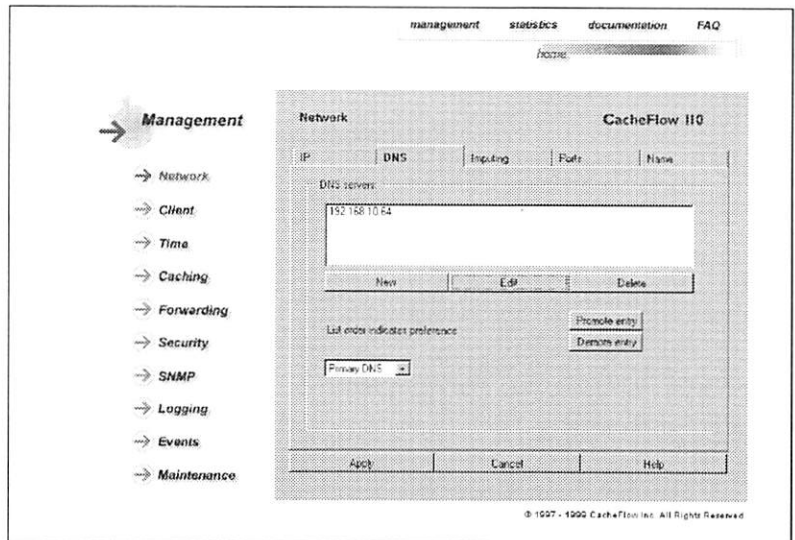


To add DNS servers

You enter the first DNS server by using the Setup Console. You can define multiple DNS servers by using the CLI or Web interface.

1. Select Management from the CacheOS home page.
2. Select the DNS tab in the Network applet.
3. To enter additional DNS servers, click the New button.
4. Type the IP address of the DNS server and click the OK button.
5. Click the Apply button to save your changes.

Figure 3-3
Adding DNS Servers

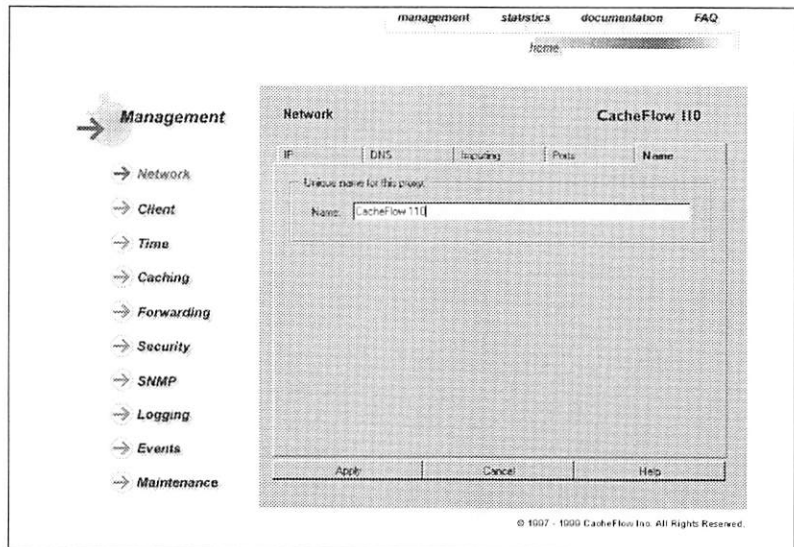


To set the CacheFlow 110 name

You can assign a name to the CacheFlow 110. The name appears in the Web interface, used for the CLI prompt. If you have entered a host name for the CacheFlow 110 in your DNS server, you should use the same name.

1. Select Management from the CacheFlow 110 home page.
2. Select the Name tab in the Network applet.
3. Enter the name in the name field.
4. Click the Apply button to save your changes.

Figure 3-4
Setting the
CacheFlow 110 Name

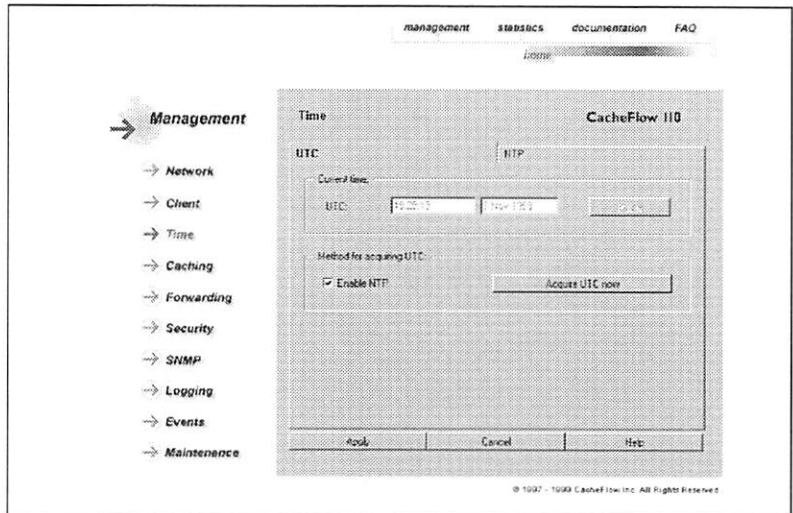


To set the UTC time

CacheOS sets UTC time by connecting to an NTP server. CacheOS includes a list of NTP servers available on the Internet. If an NTP server is not available, you can set the time manually, but you should not disable NTP, unless it is not available. To set the time manually by using the Web interface, follow these steps:

1. Select Management from the CacheFlow 110 home page.
2. Select the Time applet.
3. Clear the Enable NTP checkbox.
4. Click the Pause button.
5. Enter the current UTC time and date in the UTC fields.
6. Click the Resume button.
7. Click the Apply button to save your changes.

*Figure 3-5
Setting the Time*



Where to Go From Here

Once you install and configure the CacheFlow 110, it begins to cache Web objects. As soon as CacheFlow 110 begins running, you will see disk and network activity as it begins maintaining the cache.

Once the CacheFlow 110 becomes operational, you can configure one of the following:

- Configure your routers or load-balancing switches to forward requests transparently to the device.
- Configure the client instructions and notify users that the CacheFlow 110 is now available.

If you do not use transparent caching, you must configure each browser separately to use the CacheFlow 110. The client instructions make it easy for your users to modify their configuration. (Refer to the *CacheOS Management and Configuration Guide* for more information.)

Security Considerations

Once you configure CacheOS and it runs properly, secure the device to ensure that it is managed and used only as authorized. You can restrict access to the device using the following configuration options:

- You can limit access to the management console to specific IP addresses or subnets. To set console restrictions by using the CLI, refer to the **security** command. (To set console restrictions by using the Web interface, refer to *Setting Access Restrictions* in the *CacheOS Management and Configuration Guide*.)
- You can restrict user access to CacheOS by using the filter list ACL (access control list) options. To create an ACL using the CLI, refer to the **filter-list** command. (To create an ACL using the Web interface, refer to *Using a Filter List* in the *CacheOS Management and Configuration Guide*.)
- You can disable proxy mode to prevent unauthorized use of the cache. To turn off the proxy mode, set the proxy port to zero. To change the proxy port using the CLI, refer to the **http-proxy-port** command. (To change the proxy port using the Web interface, refer to *Setting IP Ports* in the *CacheOS Management and Configuration Guide*.)

Numerous other management options, settings and statistics exist that you can review. Refer to the *CacheOS Management and Configuration Guide* for a complete list of basic and advanced configuration procedures for using the CLI interface and Web interface.

Chapter 4

Troubleshooting

This chapter describes how to locate and solve common problems when installing and using the CacheFlow 110.

Hardware Considerations

The CacheFlow 110 arrives fully assembled as a single unit. The CacheFlow 110 does not contain any user serviceable components. If you experience problems with the proper operation of the CacheFlow 110, contact your CacheFlow Customer Service representative for assistance.

Software Considerations

Testing the CacheFlow 110

If you suspect a problem with the CacheFlow device, you can test its operation and connection using the CLI. You can use the **ping** and **traceroute** commands to test the network connection. From enable mode, you can use the **test** command to retrieve an object by using CacheOS, or run a loopback test to verify that the device is operational.

Cannot Access the Serial Console

When connecting to the Serial Console, you must configure the terminal emulator for the correct settings. You can connect to the Serial Console by using a terminal, a terminal server or a computer running a terminal emulator. For detailed instructions on connecting the terminal, go to page 10.

CLI Session Times Out

When you have a Telnet session to the CLI, that session remains open as long as activity exists. If you leave the session idle, the connection eventually times out and you have to reconnect. The default timeout is

five minutes. You can set the timeout using the **line-vty** command in configure mode.

CLI Limited to Four Sessions

Four active Telnet sessions (and one Serial Console session) limit the CLI. When you try to connect a fifth Telnet session, you will receive a message stating that no more sessions are available. To connect, you must wait for one of the active sessions to close.

Web Interface is Not Accessible

When you connect to the CacheFlow 110 Web management port from a Web browser, the Web interface should be displayed. If the Web browser fails when attempting to connect to the CacheFlow 110, follow the steps outlined below to help identify the problem:

1. Verify that you have typed the correct IP address and port (the default port is 8081) for the CacheFlow 110 in the Web browser. The only way you can verify the address the CacheFlow 110 is using without the Web interface is to connect to the Serial Console and display the network configuration. (Refer to *Connecting the CacheFlow 110 to a Terminal*, presented earlier in this guide.)
2. Verify that your workstation is configured and working properly by connecting to other Websites (such as www.cacheflow.com). If you configured your browser to use the CacheFlow 110 as the proxy server and a problem exists with the CacheFlow 110, this test may fail.
3. If you try accessing a CacheFlow 110 located on a remote network (any segment other than the segment where your workstation is attached), verify that other servers on that network are accessible.
4. Try pinging the IP address to verify that the CacheFlow 110 is accessible from the workstation. If the CacheFlow 110 does not respond to the ping, verify that it is operational, as described earlier.

Web Interface Username and Password Fail

You can protect the Web interface with a username and password. If the username and password fail, you can reset them by using the Serial Console. Go to page 18 for information on using the Serial Console.

Web Interface Applets Not Displayed

The Web interface requires Java and JavaScript. You must use a Web browser that supports Java and JavaScript, and the Web browser must have Java and JavaScript enabled. Netscape 3.0 and later versions and Internet Explorer 3.0 and later versions support Java.

If the browser encounters an error when loading an applet, restart the browser.

Web Interface Java Errors Occur

When you upgrade the Web interface, you must clear the local cache and restart the browser to guarantee that the new Java components for the Web interface are loaded. Each Web browser handles Java applets in a different way. Some browsers do not reload applets until their cache is cleared or until the browser is restarted.

User Issues

This section helps you identify common problems when the Web client either fails or is slow when retrieving Web objects.

Client HTML Requests Fail

When a request for a Web document fails, it indicates one of the following:

- A filter blocks the Web site for security or content, as when a filter blocks the source IP address or destination URL. Verify that the Website is accessible without the cache.
- The Web browser is not properly configured to use the CacheFlow 3000, as if the browser proxy settings do not point at the cache.
- The CacheFlow 3000 is not properly configured for the browser or the network. For example, the CacheFlow3000 is not configured with the correct default gateway or DNS server.
- Network problems occur, as when the Web server or the Internet connection is down.

To locate the problem, follow the steps outlined below:

1. If you use the CacheFlow 110 to access the Internet, and the CacheFlow 110 has been working properly, the most likely cause of failed requests may be the route between the

CacheFlow 110 and the Internet or intranet. Before you spend time troubleshooting the CacheFlow 110, verify that your connection to the Internet by using the ping and traceroute commands from the CLI. If you are able to ping the destination from the CacheFlow 110, use the test command to verify the device can get the object. (For information on the ping, traceroute, and test commands, refer to the *CacheOS Management and Configuration Guide*. If the request fails, try other HTML requests to verify all requests fail.)

2. The CacheFlow 110 can be configured to deny access to address groups. If the CacheFlow 110 is configured for forwarding or filtering, verify that the requested address does not match a denied subnet and mask. To configure deny settings in the Web interface, select Management from the home page and then ICP/Forwarding.
3. If your network is not configured for transparency, check the Web browser to see if it is using a PAC file for auto-configuration. If the Web browser is configured to use a PAC file, verify that the address of the PAC file is correct and that the file is accessible. (Refer to the *CacheOS Management and Configuration Guide* for additional information.) If auto-configuration is not being used, check the Web browser's proxy configuration. If you are not using transparency, you must configure the Web browser for the CacheFlow 110's IP address and port. (Refer to the *CacheOS Management and Configuration Guide* for information on configuring the clients.)
4. If the correct IP address and port for the proxy server is specified in the Web browser, try pinging the IP address to verify that the CacheFlow 110 is accessible from the workstation. If it does not respond to the ping, verify that it is operational, as described earlier. Also verify that you can ping other nodes on the network. If you can ping the CacheFlow 110, try pinging the workstation from CacheFlow 110 CLI.
5. If the CacheFlow 110 responds to the ping, verify that the CacheFlow 110's configuration is correct. If it is configured to forward requests, verify the server to which the CacheFlow 110 is forwarding requests. If the CacheFlow 110 is not forwarding requests, check the default gateway address and DNS address.
6. If the CacheFlow 110's default gateway address and DNS address are correct, try pinging each address from the CLI to verify that the servers are running. Be sure to ping the gateway

and DNS server from the same network segment where the CacheFlow 110 is connected.

7. If the default gateway is accessible, the problem most likely lies outside the local network. To verify that the problem is not associated with the CacheFlow 110, you must configure your workstation for the same gateway address as the CacheFlow 110, and configure the Web browser not to use a proxy server for HTTP requests.

Client HTML Responses are Slow

If every request from the client results in a slow response, a problem accessing a PAC file or the CacheFlow 110 can occur. To identify the problem, follow the steps outlined below:

1. Check the Web browser to see if it is using a PAC file for auto-configuration. If the Web browser is configured to use a PAC file, verify that the address of the PAC file is correct and that the file is accessible. (Refer to the *CacheOS Management and Configuration Guide* for additional information on using PAC files.) If the PAC file is not accessible, the Web browser may attempt to access the PAC file, then wait for a timeout before retrieving the object directly.
2. Check the Web browser's proxy address and port. The Web browser must be configured for the CacheFlow 110's IP address and port. (Refer to the *CacheOS Management and Configuration Guide* for information on configuring the clients.) If the address or port is not correct, the Web browser may attempt to access the CacheFlow 110, then wait for a timeout before retrieving the object directly.
3. If the correct IP address and port for the proxy server is specified in the Web browser, try pinging the IP address to verify that the CacheFlow 110 is accessible from the workstation. If the CacheFlow 110 does not respond to the ping, verify that it is operational.

Appendix A

Electrical and Environmental Specifications

CacheFlow 110 Electrical and Environment Specifications

Specifications for the CacheFlow 110 base model and a fully configured version are shown below:

	Base Model	Fully Configured
Dimensions (L x W x H)	375 x 442 x 88.1 mm 14.75 x 17.4 x 3.47 in	375 x 442 x 88.1 mm 14.75 x 17.4 x 3.47 in
Weight	9.64 kg 21.25 lb	9.64 kg 21.25 lb
Power Input	115V/230V 50 - 60 Hz	115V/230V 50 - 60 Hz
Power Consumption	100 W	100 W
Operating Temperature	41 - 122° F 5 - 50° C	41 - 122° F 5 - 50° C
Operating Humidity	< 90 % relative humidity, non-condensing	< 90 % relative humidity, non-condensing
Operating Altitude	Up to 8000 ft	Up to 8000 ft
EMC	FCC Class B EN 55022 A EN 50082-1 Iec 801-2	FCC Class B EN 55022 A EN 50082-1 Iec 801-2

	Base Model	Fully Configured
	lec 801-3 lec 801-4	lec 801-3 lec 801-4
Safety	CSA C22.2 No. 950 M95 UL 1950 3 Edition lec 950	CSA C22.2 No. 950 M95 UL 1950 3 Edition lec 950

Regulatory Statements

General

IMPORTANT: Any modifications to the unit, unless expressly approved by CacheFlow, could void the your authority to operate the equipment.

Battery Warning

CAUTION: Danger of explosion if battery is incorrectly replaced. Replace the battery only with the same or equivalent type recommended by the manufacturer. Dispose of used batteries according to the manufacturer's instructions.

Class B Digital Warning

This equipment has been tested and found to comply with the limits for a Class B digital device, pursuant to Part 15 of the FCC rules. These limits are designed to provide reasonable protection against harmful interference when the equipment is operated in a commercial environment. This equipment generates, uses and can radiate radio frequency energy, and if not installed and used in accordance with the instruction manual, may cause harmful interference to radio communications. Operation of this equipment in a residential area is likely to cause harmful interference, in which case you will be required to correct the interference at your own expense.

EMC Warning

This is a Class B product. In a domestic environment this product may cause radio interference in which case the user may be required to take adequate measures.

Declaration of Conformity



CacheFlow Inc.
650 Almaden Avenue
Sunnyvale, CA 94086

408.221.2200
408.225.2250 Fax

www.cacheflow.com

Declaration of Conformity

We, CacheFlow Inc.
650 Almaden Avenue
Sunnyvale, CA 94086 USA

Declare under our sole responsibility that the products

Cacheflow Model 110

To which this declaration relates is in conformity with the following
standards:

EN 60950: 1992
EN55022: 1994
EN50082-1: 1992
IEC 801-2
IEC 801-3
IEC 801-4

Following the provisions of the 73/23/EEC and 89/336/EEC Directives,
Including the Amending Directive 93/68/EEC

Sunnyvale, CA 94086 USA
Date 10-22-99

A handwritten signature in dark ink, appearing to read "Ray G. Myers".
Ray G. Myers
Vice President, Manufacturing
And Customer Support

Index

A

Active Caching, 4

B

baud rate, 10

C

CacheFlow 110

serial port, 10

weight, 8

CacheFlow website, 1

CacheOS, 4

client, 5

HTML requests fail, 35

HTML responses slow, 37

Console password

setting with the Serial

Console, 20

D

data bits, 10

Declaration of Conformity, 43

default gateway, 18

default gateway address

setting with the Serial

Console, 20

dimensions, 39

DNS, 29

DNS cache, 5

DNS server, 18

setting with the Serial

Console, 20

E

equipment rack, 8

rack mounting brackets, 9

warning, 8

Ethernet adapters, 24, 25, 28

Ethernet cable, 14

F

flow control, 10

H

hardware, 3

HTML requests fail, 35

HTML responses slow, 37

I

installation

connecting a terminal, 10

connecting power, 12

connecting the network, 14

mounting in an equipment

rack, 8

network settings, 18

IP address, 18, 24, 25, 28

setting with the Serial

Console, 20

L

lithium batteries, 8

M

Management Console, 17, 26
 accessing, 23
 Java applets, 35
 not accessible, 34
 password fails, 35

N

name, 25, 30
network adapters, 24, 25, 28
network connection, 14
network settings, 18
NTP, 31

O

object pipelining, 4
On/Off switch, 12

P

packing list, 7
parity, 10
pipelining, 4
power consumption, 39
power cord, 12
power supply LED, 13
power switch, 13

R

rack mounting brackets, 9

S

safety instructions, 8
Serial Console, 17, 18
 accessing, 18
 cannot access, 33
 connecting a terminal, 10
 sample session, 21
serial port, 10
setting the time, 31
specifications, 39
stop bits, 10
subnet mask, 18, 24, 25, 28
 setting with the Serial
 Console, 20

T

terminal, 10
 settings, 10
terminal emulator, 10
terminal server, 10
time, 31

U

unpacking, 7

CacheOS™

Management and Configuration Guide

Software Version, 2.1

CacheFlow®

CacheFlow Inc.

650 Almanor Avenue

Sunnyvale, California 94086

(408) 220-2200 Voice

(408) 220-2250 FAX

(888) 702-3569 Technical Support

info@cacheflow.com

www.cacheflow.com

Information contained in this document is believed to be accurate and reliable. However, CacheFlow Inc. assumes no responsibility for its use nor for any infringements of patents or other rights of third parties which may result from its use. CacheFlow Inc. reserves the right to change product specifications at any time without notice.

NOTE: This equipment has been tested and found to comply with the limits for a Class A digital device pursuant to Part 15 of the FCC Rules. These limits are designed to provide reasonable protection against harmful interference when the equipment is operated in a commercial environment. This equipment generates, uses and can radiate radio frequency energy and, if not installed and used in accordance with the instruction manual, may cause harmful interference to radio communications. Operation of this equipment in a residential area is likely to cause harmful interference in which case the user will be required to correct the interference at his own expense.

CacheFlow is a trademark of CacheFlow Inc. All other product names and services identified in this documentation are trademarks or registered trademarks of their respective companies and are used throughout this documentation in editorial fashion only for the benefit of such companies. No such use, or the use of any trade name, is intended to convey an endorsement or other affiliation with CacheFlow Inc.

Copyright 1997 – 1999 CacheFlow Inc. All rights reserved worldwide. No part of this document may be reproduced by any means nor translated to any electronic medium without the written consent of CacheFlow Inc.

Printed in USA

Document Number: 231-00407-00

Document Revision: 2.1.2

10/19/99

Contents

Introduction	1
CacheOS™ Features.....	1
The CacheOS	1
Transparent Caching	2
Using Transparency	3
Overview	3
How Transparent Caching Works	3
Limitations of Transparent Caching.....	4
Configuring your Network for Transparent Caching.....	4
Configuring Clients	5
Overview	5
Selecting Client Instructions.....	6
Using a Proxy.....	7
Using a PAC File	7
Notifying Clients	9
Client Instructions	10
Configuring Netscape Navigator 2.0 - 3.0.....	10
Configuring Netscape Communicator 4.0	11
Configuring Microsoft Internet Explorer 3.0	13
Configuring Microsoft Internet Explorer 4.0	14
Configuring Microsoft Internet Explorer 5.0	15
Configuring Hierarchical Caches	17
Introduction	17
Forwarding Options.....	17
Simple Forwarding.....	17
Advanced Forwarding	18
ICP (Internet Caching Protocol).....	19
Configuring Simple Forwarding.....	20

Configuring Advanced Forwarding	20
Advanced Forwarding Configuration Commands	20
Configuring ICP	25
ICP Configuration Commands	26
Restricting Access	27
Other Advanced Forwarding and ICP Options	30
Forwarding Order	31

Using the Command Line Interface 33

Accessing the Command Line Interface	33
Using Telnet	33
Using the Serial Console	34
Logging In	35
Entering Commands	35
Interface Modes	36
Enable Mode:	37
The Show Command	38
Configuration Parameters	38
Status Parameters	39
Using the Show Command	40
show caching	41
show configuration	42
show content-distribution	43
show download-paths	45
show dns	46
show efficiency	47
show interface	48
show ip-default-gateway	49
show ports	50
show resources	51
show snmp	52
show status	53
The Configure Command	54
Using the Configure Command	55
Clearing Configuration Settings with the NO command	55
access-log	56
bypass-list	58
caching	61
direct-deny-list	63
dns	64
event-log	65
filter-list	67
forwarding	71

ftp	72
hostname	73
http-proxy-port	74
icmp	75
interface	76
ip-default-gateway	78
line-vty	79
management-port	80
no	81
ntp	82
restart	83
return to sender	84
rip	85
security	86
snmp	87
socks-machine-id	89
static-routes	90
upgrade-path	91
WCCP	92
web-management	93
Management Commands	94
acquire-utc	95
clear-cache	96
display	97
kill	98
load	99
offline-disk	100
purge-dns-cache	101
ping	102
restart	103
restore-defaults	104
test	105
traceroute	106
upload-access-log	107

Using the Web Interface 109

Overview	109
Basic Management Tasks	110
Advanced Management Tasks	111
Statistics	111
Configuring the Network Adapters	112
Entering DNS Servers	113
Setting IP Ports	114

Setting the CacheFlow Web Cache Name	115
Selecting Client Instructions	116
Setting the Time	117
Setting the Console Username and Password	118
Setting Access Restrictions.....	119
Setting Proxy User Authentication	120
Enabling SNMP.....	121
Enabling the Access Log	122
Setting the Access Log Upload Site	123
Setting the Access Log Schedule.....	124
Setting the Access Log Format.....	125
Enabling Event Notification	126
Restarting the Device.....	127
Restoring System Defaults.....	128
Purging the DNS Cache.....	129
Clearing the System Cache	129
Upgrading the System.....	130
Displaying the Event Log	131
Using Syslog Monitoring.....	132
Using Name Imputing.....	133
Configuring Cache Options	134
Setting the Network Bandwidth Utilization.....	136
Setting Policies	137
Setting the Maximum Object Size.....	138
Caching Negative Responses	138
Guaranteed Freshness.....	138
Setting FTP Caching Options.....	138
Using ICP and Advanced Forwarding	139
Loading the Advanced Forwarding and ICP Configuration	140
Configuring WCCP Support.....	141
Using Simple Forwarding.....	142
Using a SOCKS Server.....	143
Creating a Direct or Deny List.....	145
Using a Filter List.....	146
Using the Filter List to Restrict Access	148
Creating Filter Lists.....	148
Defining Static Routes.....	151
Using a Bypass List	153
Central Bypass List	154
Using RIP	156
net.....	156
host.....	157
Parameters	157
Displaying the System Status.....	162

Displaying the Number of Objects and Bytes Served.....	163
Displaying Active Client Connections.....	164
Displaying How Resources are Currently Being Used	165
Displaying Cache Efficiency	166
Displaying the Cache Contents Distribution.....	167

Access Log Formats 169

Overview	169
Common Access Log Format	169
Squid-Compatible Log Format	170
Log Entry Types.....	171
Using a Custom Format	171

WCCP 173

Overview	173
Using WCCP and Advanced Transparent Redirection	174
WCCP Version 1.....	174
WCCP Version 2.....	174
Configuration File Syntax	178
Examples	181
Version 1 Standard HTTP Redirection	181
Version 2 Standard HTTP Redirection	182
Version 2 Standard HTTP Redirection Using a Multicast Address	183
Version 2 Standard HTTP Redirection Using a Security Password.....	183
Version 2 Reverse Proxy Service Group.....	184
Version 2 Service Group with Alternate Hashing	185

Regular Expressions 187

Overview	187
Introduction	187
Regular Expression Syntax.....	188
Regular Expression Details	190
Backslash	191
Circumflex and Dollar.....	193
Full Stop (Period, Dot).....	193
Square Brackets	194
Vertical Bar.....	195
Subpatterns.....	195
Repetition	196
Back References.....	199
Assertions.....	200

Once-Only Subpatterns	201
Conditional Subpatterns	202
Comments.....	203
Performance	204
RE ENGINE Differences From PERL	204
Regular Expression Examples	205

Index

Chapter 1

Introduction

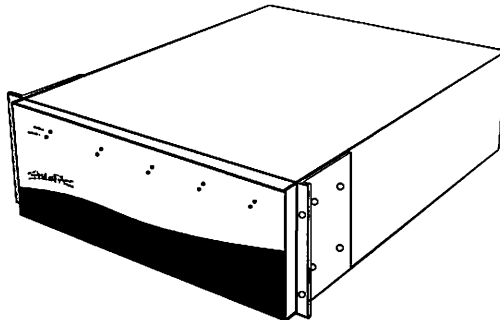
This chapter describes the CacheFlow CacheOS product features.

CacheOS™ Features

The CacheFlow Web cache provides the ultimate in Web caching performance. The Web cache is a single-purpose device, based on the patent-pending CacheOS™ operating system.

Figure 1-1
The CacheOS

The CacheOS is a single-purpose Web caching device designed specifically for caching Web objects.



The CacheOS

The CacheOS is a new generation of Web caching technology invented by CacheFlow Inc. It combines features like Active Caching, Object Prefetch, and DNS Caching with an ultra-efficient storage system.

Active Caching

Active Caching describes how the CacheOS stores and refreshes objects in the cache. When a Web page is requested by a client, the CacheOS tracks a variety of information for every object on the page, including the frequency of requests, frequency of object modifications, and time to retrieve the object. The CacheOS then uses this information to determine the refresh pattern for the object.

Rather than waiting to refresh objects as clients request them, forcing the clients to wait while the objects are verified, the CacheOS constantly analyzes the cache and refreshes objects according to each object's refresh pattern. This drastically reduces the amount of time required to deliver the objects to the clients.

Object Prefetching

Each Web page can be composed of dozens of objects, such as images, sounds, Java applets, etc. Under normal browser operations, the client requests a Web page, and the HTML document is retrieved and delivered to the client. As the client reads the HTML document, it begins requesting the rest of the objects that make up the page.

CacheFlow speeds this process. When the CacheOS retrieves the HTML document, it reads the document and begins prefetching the objects that make up the page. The CacheOS can parse the document and request the objects much faster than the client can request them. By the time the client begins requesting objects on the page, the CacheOS has already started loading them in the cache. Object prefetching delivers complete pages to the client faster, even when the page has to be retrieved from the server.

DNS Caching

The CacheOS also maintains a large DNS cache using similar active caching techniques. The shared DNS cache boosts overall Web cache performance.

Transparent Caching

The CacheOS supports transparent caching. Transparency allows you to deploy the CacheOS without requiring users to configure their browsers. This simplifies the installation, and ensures users actually use the cache. The CacheOS is configured for transparency by default. All you have to do to install the CacheFlow Web cache is connect the device to your network and configure your routers or load-balancing switches to forward all packets addressed to port 80 to the CacheOS.

Chapter 2

Using Transparency

The CacheOS supports transparent caching. Transparent caching allows you to deploy the CacheOS without modifying each user's configuration. This chapter describes the features of transparent caching, and how to configure your network.

Overview

Transparent caching makes it easy to deploy the CacheOS on your network. Instead of configuring each client to use the Web cache as an HTTP proxy, you can configure your routers or load-balancing switches to forward requests to the CacheFlow Web cache. All packets destined for port 80 of an IP address (the default HTTP port) that are sent to the CacheFlow Web cache are handled through the cache. Using transparency you can ensure all users use the cache without modifying their configuration.

How Transparent Caching Works

When a client sends a request to a Web server that is not located on the local network, that request is sent through a router or switch. Most routers or load-balancing switches allow you to filter packets based on a variety of requirements. You can filter packets by address, subnet, or IP port.

The IP port identifies a logical port at the destination address. The default port for HTTP servers is port 80. If you configure your router to forward packets addressed to port 80 to the CacheFlow Web cache, when the CacheOS receives the packets it will process the requests through the cache. If the requested object is in the cache, the CacheOS will deliver the object without requesting it from the network. If the object is not in the cache, the CacheOS will retrieve the object from the source and deliver it to the client. Because the request is forwarded from the router to the cache, the client has no knowledge of the cache.

You can use both transparent caching and proxy configuration at the same time. Clients that are configured to use the CacheOS as a proxy server are not affected by transparent caching.

Limitations of Transparent Caching

Transparent caching simplifies installation and management because you do not have to configure the clients to use a proxy server. In some environments (such as Internet Service Providers) it may be too difficult to configure clients and transparent caching will be the only viable option. In other environments, where you have greater control of the client configuration, you may want to configure the clients to take full advantage of the CacheOS (see Configuring Clients on page 5). The features supported by proxy configuration that are not supported by transparent caching are listed below:

Caching Support	Transparent	Proxy
Web servers at ports other than port 80	No	Yes
FTP caching	No	Yes
DNS caching	Yes	Yes
Filtering	Yes	Yes

Configuring your Network for Transparent Caching

To configure your network for transparent caching, all you have to do is configure your router or load-balancing switch to forward packets to the CacheFlow Web cache. Each router and switch is configured according to the manufacturer's instructions. For detailed instructions on configuring different routers and switches, go to the CacheFlow Web site and select the Technical Support section:

<http://www.cacheflow.com>

Chapter 3

Configuring Clients

If you do not configure your network to use transparent caching, each browser must be configured to use the CacheFlow Web cache. This chapter describes how to configure the client instructions on the CacheOS, and manage users. This information is only required if you do not use transparent caching as described on page 3.

Overview

If you do not use transparent caching, each client browser must be configured to use the CacheFlow Web cache as a proxy server. The CacheFlow Web cache makes it easy to configure clients, by supporting proxy auto-configuration files and including customized client instructions. The client instructions are HTML pages that use JavaScript to detect the viewing browser, and display the correct instructions. You can select the instructions that are displayed:

- Configuring the browser to use a proxy server
- Configuring the browser to use the default PAC file for automatic configuration
- Configuring the browser to use a custom PAC file for automatic configuration

NOTE: *Local caching should be disabled to take full advantage of the CacheOS.*

The instructions include the IP address of the CacheFlow Web cache, or the URL for the PAC file. They also describe how to disable local disk caching in the browser.

Once you have selected the client instructions to display, you can copy the sample email message from the Web interface to notify your users.

Selecting Client Instructions

You can select the client instructions the CacheOS displays. There are instructions for each network adapter. To select the client instructions using the Web interface, follow the steps outlined below:

1. Select Management from the CacheOS home page.
2. Select the Client applet.
3. Select the adapter you want to configure in the adapter drop-down list.
4. Select the instructions you want to display.
5. Click the Apply button to save your changes.

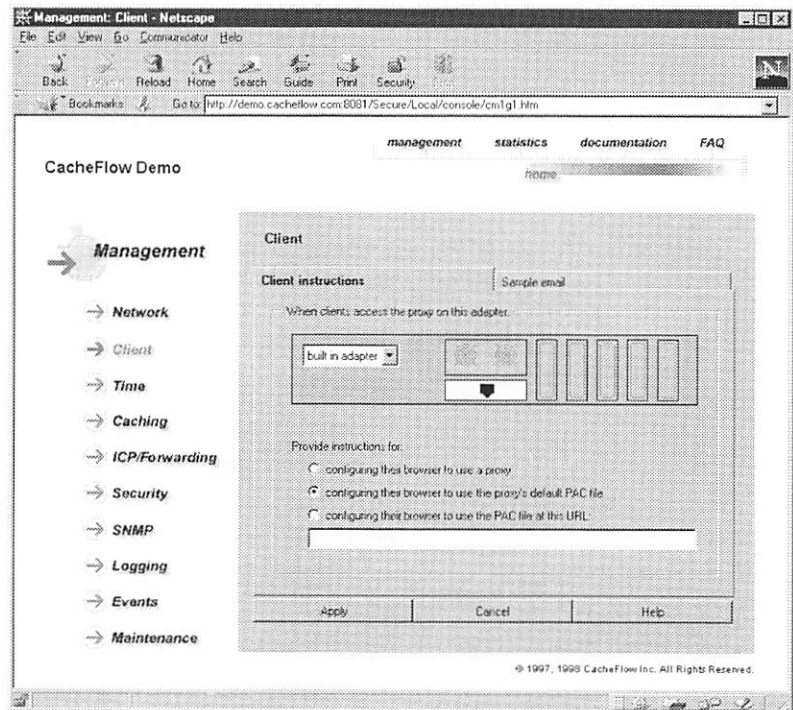


Figure 3-1
Selecting Client Instructions

You can select the instructions to display.

To set the client instructions using the command line interface, see the **interface** command on page 76.

Using a Proxy

Each browser can be configured to use a proxy server for the various Web protocols. The clients must specify the CacheFlow Web cache as the proxy server for HTTP and HTTPS requests. When they configure the browser, they must specify the IP address and port of the CacheFlow Web cache.

Using a PAC File

A PAC (proxy auto-configuration) file allows you to easily manage multiple clients. If your users are using Microsoft Internet Explorer version 3.02 or later, or Netscape version 2.0 or later, you should use a PAC file to manage the proxy configuration. When using a PAC file, the browsers need to be configured only once to point to the location of the PAC file. Proxy configuration changes can then be made without modifying each client configuration. To use a PAC file, the client must have JavaScript enabled.

The Default PAC File

The CacheOS includes a default PAC file that specifies its own IP address as the address for HTTP and HTTPS requests. You can use the default PAC file if you do not already have your own custom PAC file.

If you configure clients to use the default PAC file, you can switch to a custom PAC file in the future without changing each client's configuration. When a browser requests the default PAC file, and the CacheOS is configured to use a custom PAC file, it will return the custom PAC file.

Creating a Custom PAC File

To use your own PAC file, you must create the file and place it on a Web server available to all users.

The proxy auto-configuration file is a text file that defines a FindProxyForURL JavaScript function. When creating the file, follow the guidelines listed below:

1. Save the file with a .pac filename extension.
2. Include only the JavaScript function by itself. Do not embed the function in HTML.
3. Configure your Web server to map the .pac filename extension to the MIME type as shown below:

```
application/x-ns-proxy-autoconfig
```

If using a Netscape server, edit the mime.types file in the config directory. If using an Apache, CERN or NCSA server, use the AddType directive. If using a Microsoft server, the mime types are stored in the registry.

The FindProxyForURL Function

The FindProxyForURL function you create can be very complex, selecting the proxy based on time, date, address range, etc. Or it can be very simple. See your Web server documentation for detailed information on using PAC files. For information on using the FindProxyForURL JavaScript function, go to the CacheFlow Web site and select Technical Support from the home page:

<http://www.cacheflow.com>

FindProxyForURL Example

A simple example that sends all HTTP and HTTPS requests to a proxy server is illustrated below:

```
function FindProxyForURL(url, host) {  
    if (url.substring(0, 5) == "http:") {  
        return "PROXY 192.168.10.18:8080; DIRECT";  
    }  
    else if (url.substring(0, 6) == "https:") {  
        return "PROXY 192.168.10.18:8080; DIRECT";  
    }  
    else if (url.substring(0, 4) == "ftp:") {  
        return "PROXY 192.168.10.18:8080; DIRECT";  
    }  
    else {  
        return "DIRECT";  
    }  
}
```

You must replace the proxy server address and port in the example with the IP address of your CacheFlow Web cache.

This example sends all requests that begin with "http:", "https:" or "ftp:" to the following IP address and port: 192.168.10.18:8080. You must enter the IP address and port of your CacheFlow Web cache. For additional options that can be included in the proxy auto-configuration file, see the CacheFlow Web site.

Notifying Clients

Once you have selected the client instructions to display, you can copy the sample email message from the Web interface and send it to your users. To notify your clients, follow the steps outlined below:

1. Select Management from the CacheOS home page.
2. Select the Client applet.
3. Select the Sample email tab.
4. Copy the message text, which includes the URL where the client instructions are located.
5. Paste the message text in an email to your users, and modify the text as necessary.

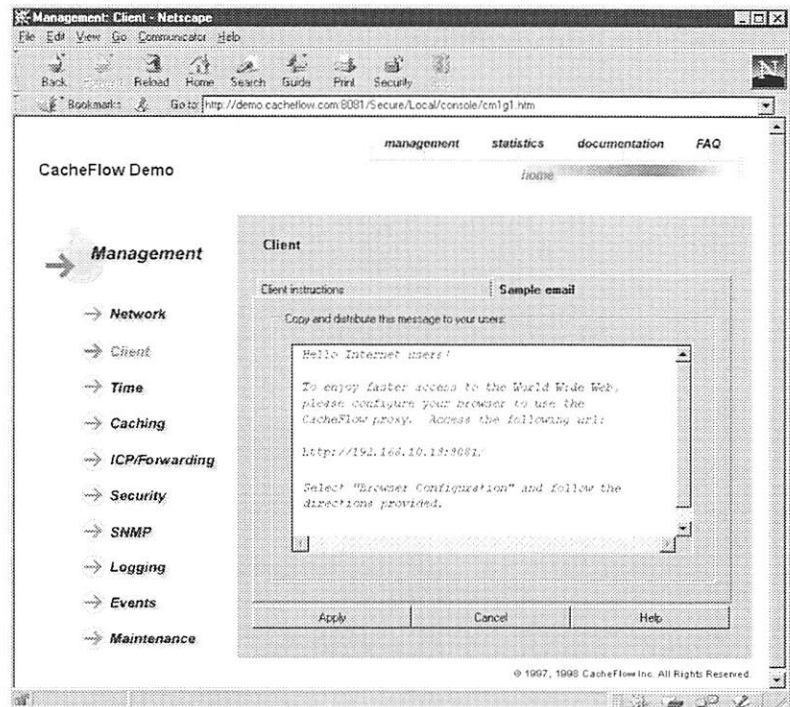


Figure 3-2

Copying the Sample Email Message

The sample email describes where the client instructions are located on the CacheFlow Web cache.

Client Instructions

When you connect to the management port of the Web cache with a Web browser, and you select Browser Configuration from the CacheOS home page, instructions are displayed that describe how to configure the browser you are using to use the Web cache. These instructions include the correct IP address of the CacheFlow Web cache, or the URL for the PAC file. Samples of these instructions are included below.

Configuring Netscape Navigator 2.0 - 3.0

Netscape Navigator supports the use of proxy servers, and PAC files.

Specifying the Proxy Server

To set the proxy server in Netscape Navigator version 2.0 or 3.0, follow the steps outlined below:

1. From the Options menu select Network Preferences.
2. Select the Proxies tab.
3. Select the Manual Proxy Configuration radio button.
4. Click the View button.
5. Type the CacheFlow Web cache IP address in the HTTP Proxy field.
6. Type the correct port for the CacheFlow Web cache in the HTTP port field (the default port is 8080).
7. Type the CacheFlow Web cache IP address in the Security field.
8. Type the correct port for the CacheFlow Web cache in the Security port field.
9. Type the CacheFlow Web cache IP address in the FTP Proxy field.
10. Type the correct port for the CacheFlow Web cache in the FTP port field (the default port is 8080).
11. Click the Ok button to save your changes.

Disabling Local Disk Caching

To disable the local disk cache used by Netscape Navigator, follow the steps outlined below:

1. From the Options menu select Network Preferences.
2. Select the Cache tab.
3. Set the Disk Cache field to zero.
4. Click the Clear Disk Cache Now button.
5. Click the Ok button to save your changes.

Using Proxy Auto-Configuration Files

Netscape Navigator version 2.0 and 3.0 support proxy auto-configuration. To use proxy auto-configuration, you must create the PAC file as described on page 7. To configure Netscape Navigator to use proxy auto-configuration, follow the steps outlined below:

1. From the View menu select Options.
2. Select the Advanced tab.
3. Click the Automatic Configuration button.
4. Enter the URL for the PAC file in the URL text box.
5. Click the Refresh button for your changes to take affect immediately.
6. Click the Ok button to save your changes.

Configuring Netscape Communicator 4.0

Netscape Communicator supports the use of proxy servers, and PAC files.

Specifying the Proxy Server

To set the proxy server in Netscape Communicator version 4.0, follow the steps outlined below:

1. From the Edit menu select Preferences.
2. Expand the Advanced category.
3. Select Proxies under the Advanced category.
4. Select the Manual Proxy Configuration radio button.
5. Click the View button.
6. Type the CacheFlow Web cache IP address in the HTTP Proxy field.

7. Type the correct port for the CacheFlow Web cache in the HTTP port field (the default port is 8080).
8. Type the CacheFlow Web cache IP address in the Security field.
9. Type the correct port for the CacheFlow Web cache in the Security port field.
10. Type the CacheFlow Web cache IP address in the FTP Proxy field.
11. Type the correct port for the CacheFlow Web cache in the FTP port field (the default port is 8080).
12. Click the Ok button to save your changes.
13. Click the Ok button to close the Preferences dialog.

Disabling Local Disk Caching

To disable the local disk cache used by Netscape Communicator, follow the steps outlined below:

1. From the Edit menu select Preferences.
2. Expand the Advanced category.
3. Select Cache under the Advanced category.
4. Set the Disk Cache field to zero.
5. Click the Clear Disk Cache button.
6. Click the Ok button to save your changes.

Using Proxy Auto-Configuration Files

Early versions of Netscape Communicator 4.0 may not work properly with PAC files.

Netscape Communicator version 4.0 supports proxy auto-configuration. To use proxy auto-configuration, you must create the PAC file as described on page 7. To configure Netscape Communicator to use proxy auto-configuration, follow the steps outlined below:

1. From the Edit menu select Preferences.
2. Expand the Advanced category.
3. Select Proxies under the Advanced category.
4. Select the Automatic Proxy Configuration radio button.
5. Enter the URL for the PAC file.
6. Click the Reload button to load the PAC file.
7. Click the Ok button to save your changes.

Configuring Microsoft Internet Explorer 3.0

Microsoft Internet Explorer version 3.0 supports the use of proxy servers, and version 3.02 supports proxy auto-configuration (PAC) files.

Specifying the Proxy Server

To set the proxy server in Microsoft Internet Explorer, follow the steps outlined below:

1. From the View menu select Options.
2. Select the Connection tab.
3. Enable the Connect through a proxy server checkbox.
4. Type the CacheFlow Web cache IP address in the HTTP proxy field.
5. Type the correct port for the CacheFlow Web cache in the HTTP port field (the default port is 8080).
6. Type the CacheFlow Web cache IP address in the Secure field.
7. Type the correct port for the CacheFlow Web cache in the Secure port field.
8. Type the CacheFlow Web cache IP address in the FTP proxy field.
9. Type the correct port for the CacheFlow Web cache in the FTP port field (the default port is 8080).
10. Click the Ok button to save your changes.

Minimizing Local Disk Caching

To minimize the local disk cache used by Microsoft Internet Explorer, follow the steps outlined below:

1. From the View menu select Options.
2. Select the Advanced tab.
3. Click the Settings button.
4. Set the Amount of disk space to use to 1% (the lowest setting).
5. Click the Ok button to save your changes.

Using Proxy Auto-Configuration Files

Microsoft Internet Explorer version 3.02 supports proxy auto-configuration. To use proxy auto-configuration, you must create the PAC file as described on page 7. To configure Microsoft Internet Explorer to use proxy auto-configuration, follow the steps outlined below:

1. From the View menu select Options.
2. Select the Advanced tab.
3. Click the Automatic Configuration button.
4. Enter the URL for the PAC file in the URL text box.
5. Click the Refresh button for your changes to take affect immediately.
6. Click the Ok button to save your changes.

Configuring Microsoft Internet Explorer 4.0

Microsoft Internet Explorer version 4.0 supports the use of proxy servers and proxy auto-configuration (PAC) files.

Specifying the Proxy Server

To set the proxy server in Microsoft Internet Explorer, follow the steps outlined below:

1. From the View menu select Internet Options.
2. Select the Connection tab.
3. Enable the Access the Internet using a proxy server checkbox.
4. Click the Advanced button.
5. Clear the Use the same proxy server for all protocols checkbox.
6. Type the CacheFlow Web cache IP address in the HTTP field.
7. Type the correct port for the CacheFlow Web cache in the HTTP port field (the default port is 8080).
8. Type the CacheFlow Web cache IP address in the Secure field.
9. Type the correct port for the CacheFlow Web cache in the Secure port field.
10. Type the CacheFlow Web cache IP address in the FTP field.
11. Type the correct port for the CacheFlow Web cache in the FTP port field (the default port is 8080).
12. Click the OK button to close the Proxy Settings dialog.
13. Click the OK button to save your changes.

Minimizing Local Disk Caching

To minimize the local disk cache used by Microsoft Internet Explorer, follow the steps outlined below:

1. From the View menu select Internet Options.
2. Click the Settings button on the General tab.
3. Set the Amount of disk space to use to 1% (the lowest setting).
4. Click the Ok button to save your changes.

Using Proxy Auto-Configuration Files

Microsoft Internet Explorer version 4.0 supports proxy auto-configuration. To use proxy auto-configuration, you must create the PAC file as described on page 7. To configure Microsoft Internet Explorer to use proxy auto-configuration, follow the steps outlined below:

1. From the View menu select Internet Options.
2. Select the Connection tab.
3. Click the Configure button.
4. Enter the URL for the PAC file in the URL text box.
5. Click the Refresh button for your changes to take affect immediately.
6. Click the Ok button to save your changes.

Configuring Microsoft Internet Explorer 5.0

Microsoft Internet Explorer version 5.0 supports the use of proxy servers and proxy auto-configuration (PAC) files.

Specifying the Proxy Server

To set the proxy server in Microsoft Internet Explorer, follow the steps outlined below:

1. From the Tools menu select Internet Options.
2. Select the Connections tab.
3. Enable the Use a proxy server checkbox.
4. Click the LAN Settings button.
5. Clear the Use the same proxy server for all protocols checkbox.
6. Type the CacheFlow Web cache IP address in the HTTP field.
7. Type the correct port for the CacheFlow Web cache in the HTTP port field (the default port is 8080).

8. Type the CacheFlow Web cache IP address in the Secure field.
9. Type the correct port for the CacheFlow Web cache in the Secure port field.
10. Type the CacheFlow Web cache IP address in the FTP field.
11. Type the correct port for the CacheFlow Web cache in the FTP port field (the default port is 8080).
12. Click the OK button to close the Proxy Settings dialog.
13. Click the OK button to save your changes.

Minimizing Local Disk Caching

To minimize the local disk cache used by Microsoft Internet Explorer version 5.0, follow the steps outlined below:

5. From the Tools menu select Internet Options.
6. Click the Settings button on the General tab.
7. Set the Amount of disk space to use to 1MB (the lowest setting).
8. Click the Ok button to save your changes.

Using Proxy Auto-Configuration Files

Microsoft Internet Explorer version 5.0 supports proxy auto-configuration. To use proxy auto-configuration, you must create the PAC file as described on page 7. To configure Microsoft Internet Explorer to use proxy auto-configuration, follow the steps outlined below:

1. From the Tools menu select Internet Options.
2. Select the Connections tab.
3. Enable the Use a proxy server checkbox.
4. Click the LAN Settings button.
5. Enable the Use automatic configuration script checkbox.
6. Enter the URL for the PAC file in the Address text box.
7. Click the OK button to close the LAN Settings dialog.
8. Click the Ok button to save your changes.



Chapter 4

Configuring Hierarchical Caches

When you have one Web cache, objects are either served from the cache or retrieved from the source. When you have multiple Web caches, you can create a cache hierarchy to manage requests. If one cache doesn't have the requested object, it can pass the request to other caches rather than going to the source. This chapter describes the options for creating and managing cache hierarchies.



Introduction

When the CacheOS receives a request for an object, it must determine how to process the request. If the object is in the cache, the CacheOS returns it. If the object is not in the cache, the CacheOS can get the object from the source or request it from another cache in the cache hierarchy.

You create a cache hierarchy by defining forwarding or ICP (Internet Caching Protocol) hosts the CacheOS can use to forward requests. A forwarding host can be any HTTP cache. An ICP host is a cache that supports the ICP protocol.

Forwarding Options

There are three forwarding options: simple forwarding, advanced forwarding, and ICP.



Simple Forwarding

Simple forwarding means the CacheOS is configured to forward all requests not found in the cache to a single forwarding gateway (or an alternate gateway if the primary gateway is unavailable). You should use simple forwarding only when you want all requests to go through the same gateway. For example, if you have one departmental cache and only

one border cache that provides access outside your network, you can use simple to forward requests to the border cache. This is illustrated in Figure 4-1 below:

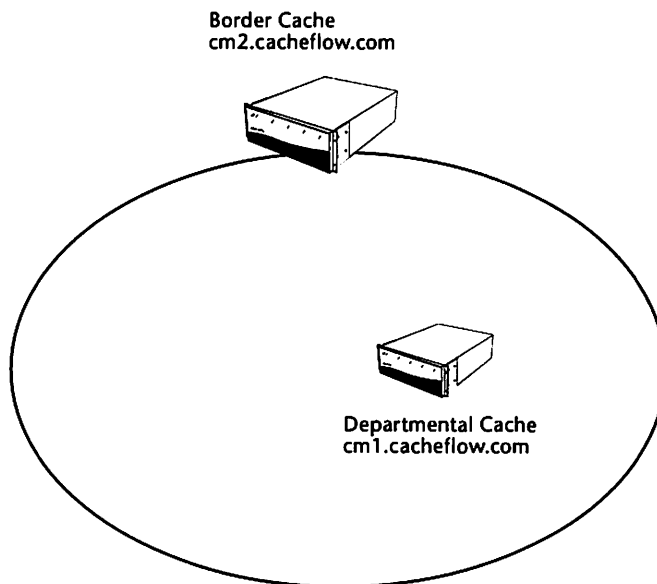


Figure 4-1
Simple Forwarding
Configuration

With simple forwarding, you can forward all requests to a single cache host gateway.

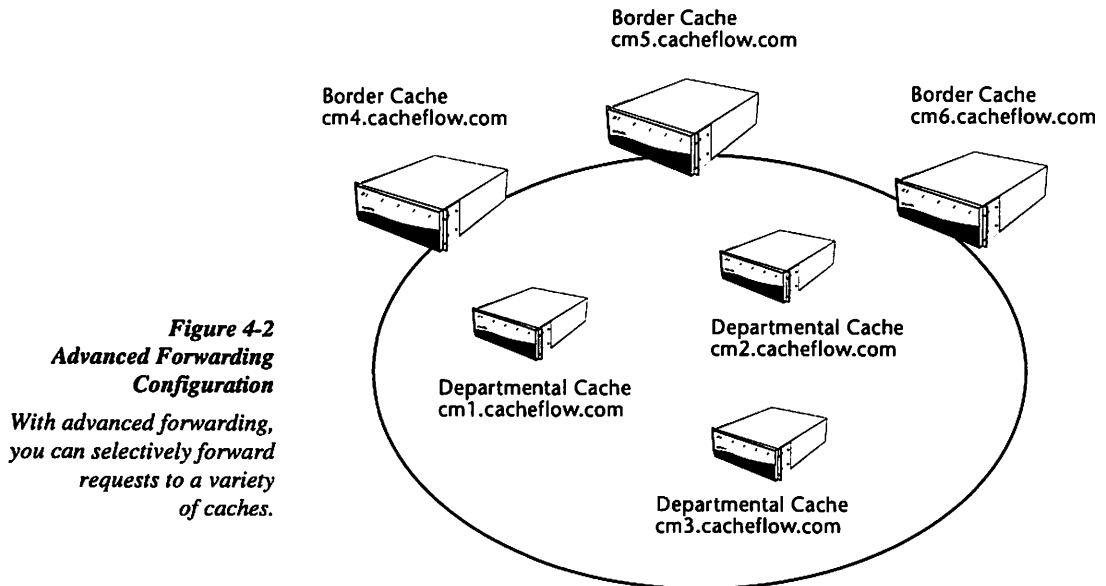
Simple forwarding forwards all requests for objects not found in the cache to the same gateway.

Advanced Forwarding

Advanced forwarding allows you to forward requests based on the URL or IP address of the requested object, and can be used to selectively forward requests to different caches, or even groups of caches. When a group of caches is defined, the CacheOS balances requests across the group, using a hash of the destination server's domain name to select the cache to use. This ensures an object from a specific server is always requested from the same cache in the hierarchy, but spreads the requests across all members of the group.

You should use advanced forwarding when there are multiple caches to which you want to forward requests, or when you want to create complex forwarding rules based on the location or type of object requested. For example, you may want to forward requests for objects on the departmental networks directly to the departmental caches, but forward requests for objects on external networks to a group of border caches.

This example is illustrated in figure 4-2.



Advanced forwarding allows you to forward different requests to specific caches.

ICP (Internet Caching Protocol)

ICP is a communication protocol for caches. It allows a cache to query other caches for an object, without actually requesting the object. By using ICP, the cache can determine if the object is available from a neighboring cache, and which cache will provide the fastest response. You should use ICP only if you have ICP hosts available for you to use, or you want the CacheOS to support ICP requests from other ICP hosts.

By design, the ICP protocol requires the requesting host to wait up to two seconds for all ICP hosts to respond to the request for an object. That means the cost of using ICP (the possibly delay just to see if the object is available) must be less than the cost of retrieving the object directly from the source. ICP is commonly used in Europe, where the cost of retrieving the data from the United States (or other countries) may be considerably higher than the potential delay of searching for the object in local ICP caches.

Configuring Simple Forwarding

To configure simple forwarding, all you have to do is define the address of a primary and alternate forwarding gateway. To define the forwarding gateways using the command line interface, see the **forwarding** command on page 71. For information on defining the forwarding gateways using the Web interface, see page 141.

Configuring Advanced Forwarding

To configure advanced forwarding you must create a configuration file and load it on the CacheOS. The advanced forwarding configuration is defined in the shared ICP/Forwarding configuration file. For information on loading the configuration using the command line interface, see the **icp** command on page 75. For information on loading the configuration using the Web interface, see page 139.

Advanced Forwarding Configuration Commands

The advanced forwarding configuration commands are listed below:

- **fwd_host**
- **fwd_host_url_regex**
- **fwd_host_domain**
- **fwd_host_ip**

The **fwd_host** command is used to specify forwarding hosts, and forwarding groups. The **fwd_host_url_regex**, **fwd_host_domain**, and **fwd_host_ip** commands define the requests to forward.

Defining a Forwarding Host

To define a single host to use as the forwarding host, use the **fwd_host** command, then enter a **fwd_host_domain**, **fwd_host_ip**, or **fwd_host_url_regex** command to forward requests to the host:

# directive	hostName	httpPort	attributes
fwd_host	cm1.cacheflow.com	8080	

# directive	hostname	domain
fwd_host_domain	cm1.cacheflow.com	eng.cacheflow.com

In this example, all requests that match the eng.cacheflow.com domain will be forwarded to cm1.cacheflow.com. Enter additional fwd_host commands to specify additional forwarding hosts. You can include the **default** attribute to define the default parent to use for requests that do not match a fwd_host_domain, fwd_host_ip, fwd_host_url_regex command.

# directive	hostName	httpPort	attributes
fwd_host	cm1.cacheflow.com	8080	
fwd_host	cm2.cacheflow.com	8080	
fwd_host	cm3.cacheflow.com	8080	default

# directive	hostname	url
fwd_host_url_regex	cm1.cacheflow.com	http://.*\cacheflow\com.*
fwd_host_url_regex	cm2.cacheflow.com	http://.*\acc\cacheflow\com.*
fwd_host_url_regex	cm3.cacheflow.com	http://.*\ec\cacheflow\com.*

In this example, requests will be forwarded as follows:

- all requests for the acc.cacheflow.com domain will be forwarded to cm2.cacheflow.com
- all requests for the ec.cacheflow.com domain will be forwarded to cm3.cacheflow.com
- all other requests to the cacheflow.com domain will be forwarded to cm1.cacheflow.com
- all requests that do not match a fwd_host_domain, fwd_host_url_regex, or fwd_host_ip specification will be forwarded to cm3.cacheflow.com

Defining a Forwarding Group

If you want to balance requests over a group of forwarding hosts, you can define forwarding groups. To define a forwarding group, specify the group attribute with the fwd_host command, then specify the group name

with the `fwd_host_domain`, `fwd_host_ip`, or `fwd_host_url_regex` command:

```
# directive  hostname  httpPort  attributes
fwd_host    cm4.cacheflow.com  8080      group=border1
fwd_host    cm5.cacheflow.com  8080      group=border1
fwd_host    cm6.cacheflow.com  8080      group=border1

# directive      groupname      domain
fwd_host_domain  border1        company.com

# directive      groupname      IP address/subnet mask
fwd_host_ip      border1        38.160.150.0/255.255.255.0
```

In this example, all requests for the `company.com` domain, or an IP address that matches the subnet specification, will be sent to the `border1` group.

The fwd_host Command

The `fwd_host` command defines forwarding hosts to use with the `fwd_host_domain`, `fwd_host_ip`, and `fwd_host_url_regex` commands. The parameters for the `fwd_host` command are described below:

`fwd_host hostname httpPort [default | backup | group=groupname]`

Parameters:	Value	Description
hostname		The host name of the cache.
HTTPport		TCP port where the cache accepts HTTP requests. The common HTTP port is 8080.

Parameters:	Value	Description
default	groupname	If specified, designates a cache host to be the default forwarding host. All requests that do not match a fwd_host_domain, fwd_host_ip, or fwd_host_url_regex specification will be forwarded to the default host.
backup		A backup default cache host if the default host is not available.
group		The forwarding group to which this cache host belongs. When you define a forwarding group, the CacheOS will balance requests across members of the group. The first instance of the group name creates the group.

The fwd_host_domain Command

The fwd_host_domain command defines which requests are sent to which cache hosts or cache host groups. The parameters for the fwd_host_domain command are described below:

fwd_host_domain hostname[groupname] domain

Parameters:	Description
hostname[groupname]	Either the host name or group name defined by a fwd_host command. You can also handle requests locally by specifying "direct" for the host name, or deny requests by specifying "deny" for the host name.
domain	The domain to match. All requests that match the specified domain will be forwarded to the cache host or group. If you specify an asterisk (*) for the domain, the host or group will be used for all requests that fail to match a domain specification.

The fwd_host_ip Command

The fwd_host_ip command works like the fwd_host_domain command, except you can specify an IP address and subnet mask rather than a domain. If you use the fwd_host_ip command, the CacheOS will look up the IP address for requests that specify a domain name, using DNS. The DNS requests are cached, but the lookups may still affect performance.

The parameters for the fwd_host_ip command are described below:

fwd_host_ip hostname|groupname IP address/subnet mask

Parameters:	Description
hostname groupname	Either the host name or group name defined by a fwd_host command. You can also handle requests locally by specifying "direct" for the host name, or deny requests by specifying "deny" for the host name.
IP address/subnet mask	The IP address to match. All requests that match the specified address and subnet mask will be forwarded to the cache host. If you specify 0.0.0.0 for the address, the host or group will be used for all requests that fail to match an address specification.

The fwd_host_url_regex Command

The `fwd_host_url_regex` command works like the `fwd_host_domain` command, except you can specify a regular expression to match the URL. For information on using regular expressions, see "Regular Expressions" on page 173. The parameters for the `fwd_host_url_regex` command are described below:

`fwd_host_url_regex hostname|groupname URL`

Parameters:	Description
hostname groupname	Either the host name or group name defined by a <code>fwd_host</code> command. You can also handle requests locally by specifying "direct" for the host name, or deny requests by specifying "deny" for the host name.
URL	The URL specification to match. The URL specification will be treated as a regular expression string. All requests that match the URL will be forwarded to the cache host.

Configuring ICP

An ICP cache hierarchy is comprised of a group of caches with defined parent and sibling relationships. A relationship exists between two caches. A cache parent is a cache that can return the object if it is in the cache, or request the object from the source on behalf of the requester if the object is not in the cache. A cache sibling is a cache that can only return the object if it is in the cache. One cache acting as a parent can also act as a sibling to other caches.

The ICP conversation between caches is simple:

1. When an object is not in the cache, the cache sends an ICP query to its neighbors (parents and siblings) to see if any of its peers holds the object.
2. Each neighbor that holds the requested object returns an ICP_HIT reply.

3. Each neighbor that does not hold the object returns an ICP_MISS reply.

Based on the responses, the cache can determine where to request the object: from one of its neighbors, or from the source. If an ICP_HIT reply is received, the request is sent to the cache host that returned the first reply. If no ICP_HIT reply is received, the request is forwarded to the first parent that replied. If no parents respond or are configured, the request is made directly to the source.

ICP Configuration Commands

To configure ICP you must create a configuration file and load it on the CacheOS. The ICP configuration is defined in the shared ICP/Forwarding configuration file. For information on loading the configuration using the command line interface, see the **icp** command on page 75. For information on loading the configuration using the Web interface, see page 139.

The ICP commands are listed below:

```
icp_host hostname peertype HTTPport ICPport [default|backup]
icp_access_domain allow|deny domain
icp_access_ip allow|deny IP address/subnet mask
```

icp_host

The *icp_host* command describes cache peers in the hierarchy. There should be one entry for each cache you want to use. The parameters for the *icp_host* command are described below:

```
icp_host hostname peertype HTTPport ICPport [default | backup | feeder]
```

Parameters:	Value	Description
hostname		The host name of the cache.
peertype	parent sibling	Relationship of the cache to the cache you are configuring.
HTTPport		TCP port where the cache accepts HTTP requests. The common HTTP port is 8080.

ICPport		UDP port where the cache accepts ICP requests. The common ICP port is 3130.
default		If specified, designates a cache host parent to be the default ICP parent. If no ICP reply is received, all requests will be forwarded to the default parent.
backup		If specified, designates a cache host parent to be the backup default ICP parent. If the default parent is not available, the CacheOS will use the backup default parent.
feeder		If specified, designates the cache host sibling as a feeder-type host, using ICP request loops to populate the cache.

Sample icp_host commands are listed below:

```
# Define ICP parent and sibling caches.
icp_host cm1.cacheflow.com parent 8080 3130 default
icp_host cm2.cacheflow.com sibling 8080 3130
icp_host cm3.cacheflow.com sibling 8080 3130
icp_host cm4.cacheflow.com sibling 8080 3130
icp_host cm5.cacheflow.com parent 8080 3130
```

Restricting Access

You can restrict access to the CacheOS Web cache by other ICP hosts using the `icp_access_domain` and `icp_access_ip` commands. By default, when ICP is configured all ICP hosts are allowed access. You should deny access to all domains other than the ICP hosts you want to use.

The `icp_access_domain` command requires a reverse DNS lookup of each ICP query to validate the IP address. Whenever possible, you should use the `icp_access_ip` command to avoid potential problems.

`icp_access_domain`

The `icp_access_domain` command defines which hosts can request objects from the Web cache using ICP. The default action is to allow all requests. When you use `icp_access_domain`, each ICP query requires a reverse DNS lookup to validate the IP address. Depending on the number of ICP requests, these lookups can consume CacheOS resources.

The parameters for the `icp_access_domain` command are described below:

`icp_access_domain allow/deny domain`

Parameters:	Description
<code>allow/deny</code>	Allow or deny ICP queries from neighbors that match the domain specification.
<code>domain</code>	The domain to match. All ICP queries from neighbors that match the specified domain will be handled by the cache host. The special domain of "all" defines the default action when there is no domain match.

Sample `icp_access _domain` commands are listed below:

```
# allow ICP access to this CacheFlow Web cache from the cacheflow.com domain
icp_access_domain allow cacheflow.com
icp_access_domain deny all
# the deny all option should always be specified to deny all other domains
```

icp_access_ip

The `icp_access_ip` command works like the `icp_access_domain` command, except you can specify an IP address and subnet mask rather than a domain. The parameters for the `icp_access_ip` command are described below:

`icp_access_ip allowdeny address/subnet`

Parameters:	Description
allowdeny	Allow or deny ICP queries from neighbors that match the address specification.
address/subnet mask	The address and subnet mask to match. All ICP queries that match the specified address will be handled by the cache host. The special address of 0.0.0.0 defines the default action when there is no address match.

Sample `icp_access_ip` commands are listed below:

```
# allow ICP access to this CacheFlow Web cache from the local subnet
icp_access_ip allow 192.168.10.0/255.255.255.0
icp_access_ip deny 0.0.0.0
# the deny all option should always be specified to deny all other domains
```

Other Advanced Forwarding and ICP Options

In addition to the ICP and advanced forwarding commands described in the sections above, you can specify the following commands in the ICP/Forwarding configuration file:

```
icp_port 0
neighbor_timeout 2
icp_failcount attempts
http_failcount attempts
host_fail_notify onoff
host_recover_notify onoff
```

icp_port

The `icp_port` command sets the port the CacheOS will use to listen for ICP requests. The default port is 3130. If you set the port to 0, ICP will be disabled.

neighbor_timeout

The `neighbor_timeout` command sets the number of seconds the CacheOS will wait for ICP replies. When the CacheOS sends out an ICP request, it will wait for either all hosts to reply or for the `neighbor_timeout` to expire. The default timeout is set to 2 seconds.

icp_failcount

The `icp_failcount` command sets the number of consecutive failures the CacheOS can receive before considering the ICP host as failed. By default, the ICP failure count is set to 20. Each time a request fails, the failure count is incremented. When a request succeeds, the failure count is reset to zero.

http_failcount

The `http_failcount` command sets the number of consecutive failures the CacheOS can receive before considering the HTTP host as failed. By default, the HTTP failure count is set to 5. Each time a request fails, the failure count is incremented. When a request succeeds, the failure count is reset to zero. When an HTTP host fails, the CacheOS will wait five minutes before attempting to use it again as a forwarding target. If the next request fails, the CacheOS will continue to wait five minutes between attempts until the cache becomes available.

host_fail_notify

The `host_fail_notify` command tells the CacheOS to send an event notification email when a connect attempt fails persistently.

host_recover_notify

The `host_recover_notify` command tells the CacheOS to send an event notification email when a failed host recovers.

Forwarding Order

To determine how a request should be handled, the CacheOS checks for a forwarding match in the following order:

1. Check for an advanced forwarding regular expression match on the URL (`fwd_host_url_regex`). If there are multiple regular expressions that match the URL, the longest expression is used.
2. Check for an advanced forwarding domain name match on the request (`fwd_host_domain`).
3. Check for an advanced forwarding IP address match on the request (`fwd_host_ip`).
4. Check for an advanced forwarding default domain match (`fwd_host_domain` with a domain of "*").
5. If there are ICP peers, send out ICP queries and wait for a response or timeout.
6. Check for a healthy advanced forwarding default host (`fwd_host`).
7. Check for a healthy advanced forwarding backup host (`fwd_host`).
8. Check for a simple forwarding direct or deny list match.
9. Check for a healthy primary gateway for simple forwarding.
10. Check for a healthy alternate gateway for simple forwarding.
11. Do not forward the request, retrieve the object from the source.

The CacheOS uses the first match to process the request.

Chapter 5

Using the Command Line Interface

The CacheOS™ command line interface allows you to configure and manage the CacheFlow Web cache using Telnet or the Serial Console interface.

Accessing the Command Line Interface

To manage the CacheFlow Web cache you can use the Web interface, or the command line interface. The command line interface is a powerful command-based shell that allows you to monitor and view the Web cache. You can use the command line interface to view the configuration and statistics, backup and restore the configuration, and change the configuration. You can access the command line interface using Telnet or by connecting to the Serial Console.

Using Telnet

You can access the command line interface using Telnet or the Serial Console interface. To use Telnet, simply open a connection to the IP address of the Web cache (the Web cache listens for connections on the default Telnet port of 23):

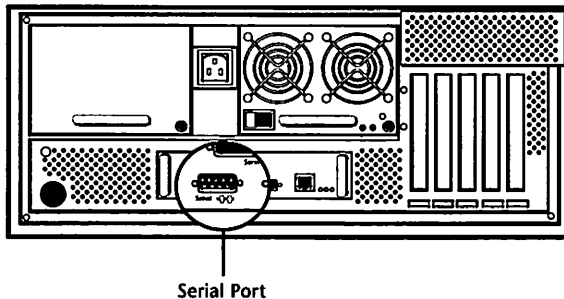
```
telnet> open 192.168.10.18<Enter>
username: admin<Enter>
password: *****<Enter>
CacheOS >
```

Using the Serial Console

To access the Serial Console, connect a terminal to the Web cache serial port. You can use any standard serial (RS232) terminal. The terminal can be a direct connected terminal, a terminal connected through a terminal server, or a computer running a terminal emulator.

Figure 5-1
Serial Connection

*CacheFlow 1000 shown.
Use the null modem cable
supplied with the Web cache
to connect a terminal.*



The terminal must be configured for the following settings:

- Baud rate: 9600 bps
- Data bits: 8
- Parity: none
- Stop bits: 1
- Flow control: none
- Smooth-scroll disabled

Once the terminal is connected to the serial port, press <Enter> three times to display the console prompt:

Welcome to the CacheOS Serial Port Console

v2.0.00 Release id: 11407

----- MENU -----

- 1) Command Line Interface
 - 2) Setup Console
-

Enter option:

At the command prompt, enter a one (1) to start the command line interface. Enter a two (2) to start the setup console, an interactive configuration wizard that allows you to configure the basic settings of the

Web cache. The setup console is only available when connected through the Serial Console.

```
----- MENU -----  
1) Command Line Interface  
2) Setup Console  
-----
```

Enter option: **1<Enter>**

Welcome to the CacheOS command line interface

Type "exit" at the main prompt to quit
CacheOS >

Logging In

When you connect to the command line interface, you are prompted for the console username and password configured when you installed the Web cache. If you have forgotten the username or password, you can reset them using the setup console.

The command line interface uses two passwords: the CacheOS console password is required to establish a connection to the interface, and an enable password can be set to restrict access to the enable mode configuration options. The console and enable passwords are set initially using the setup console.

Entering Commands

To display the list of commands available at the current prompt, type a question mark (?):

```
CacheOS > ?  
disable  
display  
enable  
exit  
help  
ping  
show  
traceroute  
CacheOS >
```

You can type the complete command, or the first two or more characters that make the command unique. For example, you can enter **sh** or **show** for the **show** command. You can also press the <Tab> key to complete the command. If you enter a question mark while typing the command, the complete command is displayed:

```
CacheOS > sh?  
show  
CacheOS > sh
```

If the command requires parameters, you can enter the parameters or enter a space and a question mark to display the parameters required:

```
CacheOS > display ?  
<url>  
CacheOS > display
```

If no parameters are required, <cr> is displayed to signify a carriage return.

Interface Modes

The command line interface has two modes: standard mode, and enable mode. The standard mode commands allow you to view the configuration settings. The enable mode commands allow you to change the configuration. The commands available for each mode are listed below:

Standard Mode:

Command	Description
disable	Turn off privileged commands
display	Display a text-based URL
enable	Turn on privileged commands
exit	Exit command line interface
help	Information on displaying help
ping	Send echo messages
show	Show running system information
tracert	Trace route to destination

Enable Mode:

Command	Description
acquire-utc	Acquire UTC from NTP server
clear-cache	Clear the contents of the cache
configure	Enter configuration mode
disable	Turn off privileged commands
display	Display a text-based URL
enable	Turn on privileged commands
exit	Exit command line interface
help	Information on displaying help
kill	Terminate a telnet session
load	Load an installable list
offline-disk	Take a disk offline
ping	Send echo messages
purge-dns-cache	Clear the DNS cache
restart	Restart the Web cache
restore-defaults	Restore the default configuration
show	Show running system information
static-route	Set static routing configuration
tracert	Trace route to destination
upload-access-log	Upload the access log to configured host

When you connect to the command line interface, you are in standard mode. To enter enabled mode, type the **enable** command and enter the enable password (if required) when prompted:

```
telnet> open 192.168.10.18<Enter>
username: admin<Enter>
password: *****<Enter>
CacheOS > enable<Enter>
password: *****<Enter>
CacheOS #
```

When you enter enabled mode, the prompt changes to a pound (#) character. To leave enabled mode, type exit or press Ctrl-Z.

The Show Command

You can use the **show** command from the standard or enabled mode of the command line interface. The **show** command displays information on the current configuration and statistics.

Configuration Parameters

The configuration parameters of the **show** command are listed below:

Parameter	Description
access-log	Access log settings
bypass-list	Bypass list
caching	Caching settings
configuration	Current configuration that differs from the default settings
direct-deny-list	Direct or deny list
download-paths	Downloaded configuration paths
dns	DNS servers and name imputing
event-log	Event log setting
filter-list	Current filter list

Parameter	Description
forwarding	Forwarding settings
hostname	Hostname
icp-settings	ICP settings
interface	Interface status and configuration
ip-default-gateway	Default IP gateway
ip-route-table	Route table information
ntp	NTP servers and information
ports	HTTP and console port
restart	System restart settings
rip	RIP settings
security	Displays the current security settings. This option is only available in enable mode.
sessions	Information on Telnet sessions
socks-machine-id	Machine id for SOCKS
snmp	SNMP configuration
static-routes	Static route table information
terminal	Terminal configuration parameters
web-management	Web management status

Status Parameters

You can use the **show** command to display the status and statistics of the Web cache. The status parameters of the **show** command are listed below:

Parameter	Description
clock	Current time
content-distribution	Sizes of objects in the cache
cpu	CPU usage
disk	Disk status and information
efficiency	Efficiency statistics
http-stats	HTTP statistics
ip-stats	TCP/IP statistics

Parameter	Description
resources	Allocation of system resources
status	Current system status
version	System hardware and software status

Using the Show Command

To use the **show** command, enter the parameter you want to display. For example, to display the current DNS configuration settings, enter the following show command:

```
CacheOS > show dns<Enter>
DNS servers:
  192.168.10.64
Imputed names:
```

Commonly used **show** commands are described below. For syntax help on commands not included here, use the command line interface help.

show caching

Displays the current configuration settings for the cache.

Syntax

show caching

The **show caching** command has no keywords or parameters.

Example

CacheOS > **show caching**<Enter>

Refresh:

Desired access freshness is 99.0%

Let CacheOS manage refresh bandwidth

Policies:

Do not cache objects larger than 50 megabytes

Cache negative responses for 0 minutes

Let CacheOS manage freshness

FTP caching:

Caching FTP objects is enabled

Do not cache FTP objects larger than 50 megabytes

FTP objects with last modified date, cached for 10% of last modified time

FTP objects without last modified date, initially cached for 24 hour

show configuration

Displays the current CacheOS configuration as it differs from the default settings. You can capture the output of this command to a text file for future reference, or to restore the configuration using the **configure** command.

Syntax

show configuration

The **show configuration** command has no keywords or parameters.

You can use the **show configuration** command to display the current configuration, then save the output to a file that can later be used to restore the configuration. When you enter the **configure** command from enable mode, the command line interface prompts you to configure from a terminal or the network. When you enter **network** you can specify the URL where the configuration file is located. The output of the **show configuration** file can be used with the **configure** command to restore the configuration.

show content-distribution

Displays the number of objects in the cache by size.

Syntax

show content-distribution

The **show content-distribution** command has no keywords or parameters.

Example

```
CacheOS > show content-distribution<Enter>
Object distribution, objects smaller than 10Kb
0Kb - 1Kb: 29702
1Kb - 2Kb: 25093
2Kb - 3Kb: 16181
3Kb - 4Kb: 9292
4Kb - 5Kb: 6638
5Kb - 6Kb: 5318
6Kb - 7Kb: 3655
7Kb - 8Kb: 3461
8Kb - 9Kb: 3192
9Kb - 10Kb: 2902
Object distribution, objects between 10Kb and 100Kb
10Kb - 20Kb: 13925
20Kb - 30Kb: 2931
30Kb - 40Kb: 1140
40Kb - 50Kb: 599
50Kb - 60Kb: 503
60Kb - 70Kb: 330
70Kb - 80Kb: 228
80Kb - 90Kb: 108
90Kb - 100Kb: 67
Object distribution, objects between 100Kb and 1Mb
100Kb - 200Kb: 186
200Kb - 300Kb: 42
300Kb - 400Kb: 24
400Kb - 500Kb: 14
500Kb - 600Kb: 16
600Kb - 700Kb: 4
700Kb - 800Kb: 12
800Kb - 900Kb: 1
900Kb - 1Mb: 1
Object distribution, objects larger than 1Mb
1Mb - 2Mb: 12
2Mb - 3Mb: 14
3Mb - 4Mb: 16
4Mb - 5Mb: 19
5Mb - 6Mb: 1
6Mb - 7Mb: 3
7Mb - 8Mb: 1
```

8Mb - 9Mb: 2
9Mb - 10Mb: 1
10Mb - 20Mb: 3
20Mb - 30Mb: 3
30Mb - 40Mb: 0
40Mb - 50Mb: 0
exceeding 50Mb: 0

show download-paths

Displays the location of the various configuration files used by the Web cache.

Syntax

show download-paths

The **show download-paths** command has no keywords or parameters.

Example

CacheOS > **show download-paths**<Enter>

Filter list

Local filter list: <http://sys.cacheflow.com/files/filter.txt>

Central filter list: <http://sys.cacheflow.com/support/subscriptions/CentralFilterList.txt>

Update when changed: no

Notify when changed: yes

Bypass list

Local bypass list: <http://sys.cacheflow.com/files/bypass.txt>

Central bypass list: <http://sys.cacheflow.com/support/subscriptions/CentralBypassList.txt>

Update when changed: no

Notify when changed: yes

Static route table: <http://sys.cacheflow.com/files/routes.txt>

ICP settings: <http://sys.cacheflow.com/files/icplist.txt>

Direct Deny list: <http://sys.cacheflow.com/files/dodlist.txt>

RIP settings: <http://sys.cacheflow.com/files/rip.txt>

Upgrade image: http://sys.cacheflow.com/files/cf_1000.chk

show dns

Displays the current DNS configuration for the Web cache.

Syntax

show dns

The **show dns** command has no keywords or parameters.

Example

CacheOS > **show dns**<Enter>

DNS servers:

192.168.10.64

Imputed names:

cacheflow.com

show efficiency

Displays statistics on objects handled by the Web cache.

Syntax

show efficiency

The **show efficiency** command has no keywords or parameters.

Example

```
CacheOS > show efficiency<Enter>
Efficiency (by objects):
  Served from cache:      81256 (46%)
  Retrieved from source:  16262 (9%)
  Noncacheable:          76737 (43%)
  Verified fresh:        2121 (1%)
Efficiency (by bytes):
  Served from cache:      327987315 (56%)
  Retrieved from source:  187995105 (32%)
  Noncacheable:          57571087 (9%)
  Verified fresh:        10832879 (1%)
Noncacheable object breakdown:
  Pragma no-cache:       15142
  Password provided:     24139
  Data in request:       0
  Not a GET request:     1231
  Cookie in response:    7627
  Password required:     0
  Negative response:     21271
  Client unique CGI:     7327
Access pattern:
  Accessed from RAM:     502337536 (92%)
  Accessed from disk:    41197568 (7%)
```

show interface

Displays the status and configuration of the network interfaces.

Syntax

show interface #all

The number of the interface to show, from 0 to the number of interfaces installed.

all Display all interfaces.

Example

```
CacheOS > show interface 0<Enter>
Ethernet interface 0
  Internet address: 192.168.10.18
  Internet subnet: 255.255.255.0
  Interface autosensed full duplex, 100 Mb/s network
  Inbound connections will be accepted by this interface
```

show ip-default-gateway

Displays the default IP gateway.

Syntax

show ip-default-gateway

The **show ip-default-gateway** command has no keywords or parameters.

Example

```
CacheOS > show ip-default-gateway<Enter>  
Default IP gateway: 192.168.10.72
```

show ports

Displays the HTTP and console port for the Web cache.

Syntax

show ports

The **show ports** command has no keywords or parameters.

Example

CacheOS > **show ports**<Enter>

HTTP: 8080

Transparent: 80

Console: 8081

show resources

Displays the allocation of system resources.

Syntax

show resources

The **show resources** command has no keywords or parameters.

Example

CacheOS > **show resources**<Enter>

Disk resources:

Available to cache: 10454155264

In use by cache: 42563354624

In use by system: 1331188736

In use by access log:281411584

Total disk installed: 54630110208

Memory resources:

In use by cache: 915042304

In use by system: 51141128

In use by network: 107558392

Total RAM installed: 1073741824

show snmp

Displays the current SNMP configuration.

Syntax

show snmp

The **show snmp** command has no keywords or parameters.

Example

CacheOS > **show snmp**<Enter>

General info:

SNMP is disabled

MIB variables:

sysContact:

sysLocation:

Community strings:

Read community: public

Write community: public

Trap community: public

Traps:

Trap address 1: 192.168.10.82

Trap address 2:

Trap address 3:

Authorization traps: disabled

show status

Displays the current system status.

Syntax

show status

The **show status** command has no keywords or parameters.

Example

CacheOS > **show status**<Enter>

Configuration:

Disks installed:	4
Memory installed:	1024 megabytes
Software version:	2.0.3
Release id:	11407
Machine id:	00A0C960217A

General status:

Last access log upload:	log has never been uploaded
Current access log size:	0
System started:	Tue, 09 Feb 1999 16:50:03 GMT
CPU utilization:	6%

The Configure Command

You can use the **configure** command from enabled mode only. The **configure** command allows you to modify the following configuration settings:

Command	Description
access-log	Configure access log
bypass-list	Installation parameters for bypass list
caching	Modify caching parameters
direct-deny-list	Installation parameters for direct or deny list
dns	Modify DNS settings
event-log	Configure event log parameters
filter-list	Installation parameters for filter list
forwarding	Forward requests to another cache
hostname	Set system hostname
http-proxy-port	Specify proxy port for HTTP requests
icp	Installation parameters for ICP settings
interface	Select an interface to configure
ip-default-gateway	Specify the default IP gateway
line-vty	Configure a terminal line
management-port	Specify port for WEB console
no	Clear certain parameters
ntp	Modify NTP parameters
restart	System restart behavior
rip	Modify RIP configuration.
security	Modify security parameters
snmp	Modify SNMP parameters
socks-machine-id	Specify machine id for SOCKS
static-routes	Installation parameters for static routes table
transparent-proxy	Enable or disable transparent proxy
upgrade-path	Network path to download system software
web-management	Enable or disable WEB console

Using the Configure Command

The **configure** command allows you to configure CacheOS settings from your current terminal session, or by loading a text file of commands from the network. The syntax for the **configure** command is shown below:

configure terminal|network <url>

When you enter the **configure** command without specifying the terminal or network parameter, you are prompted as shown below:

```
CacheOS # configure<Enter>
Configuring from terminal or network [terminal]? <Enter>
CacheOS #(config)
```

To use the **configure** command to load a configuration file, specify the URL where the file is located as shown below:

```
CacheOS # configure network sys.cacheflow.com/files/config.txt<Enter>
```

When you configure the CacheOS from the terminal, the prompt changes to **#(config)**. At the config prompt, you can enter any of the configure commands. For a list of available commands when using the command line interface, enter a question mark. To exit configure mode, press Ctrl-Z or enter the **exit** command.

Clearing Configuration Settings with the NO command

To clear the configuration for some settings, use the **no** command. The parameters supported by the **no** command are listed below:

Parameter	Description
ip-default-gateway	Set the default ip gateway to zero.
socks-machine-id	Remove the SOCKS machine id.
upgrade-path	Clear the upgrade image download path.

The commonly used **configure** commands are described below. For syntax help on commands not included here, use the command line interface help.

access-log

Allows you to configure the access log settings.

The CacheOS can maintain an access log for each HTTP request made. The access log can be stored in one of three formats, which can be read by a variety of reporting utilities (see "Access Log Formats" on page 169 for additional information on log formats).

Syntax

access-log

When you enter the **access-log** command, the interface displays the config access-log prompt, where you can enter access-log commands:

```
CacheOS #(config) access-log<Enter>
CacheOS #(config access-log)
```

The access-log commands are listed below:

Command	Parameters	Description
action	stopupload	What to do if access log exceeds allotted size.
alternate	host <host name> password <password> path <path> username <username>	Configure secondary access log upload site.
disable		Disable access logging.
enable		Enable access logging.
exit		Return to config prompt.
filename-prefix	<prefix string>	Configure upload filename prefix
format	common squid-compatible custom	Configure access log format.
no	alternate primary	Delete primary or alternate upload site.

Command	Parameters	Description
primary	host <host name> password <password> path <path> username <username>	Configure primary access log upload site.
upload	daily hourly	Specify access log upload interval.
threshold	<percent>	Percent of disk access log can consume.

Example

```
CacheOS #(config) access-log<Enter>
CacheOS #(config access-log) enable<Enter>
ok
CacheOS #(config access-log) format squid-compatible<Enter>
ok
CacheOS #(config access-log)
```

bypass-list

Sets the bypass list options.

The bypass list is only used for transparent caching.

Bypass routes are used to prevent the CacheFlow device from transparently proxying requests to servers that perform IP authentication with clients. The bypass list contains a list of IP addresses, subnet masks, and gateways. When a request matches an IP address and subnet mask specification in the bypass list, the request is sent to the designated gateway.

To use bypass routes, you must create a text file that contains a list of address specifications. The file should be named with a .txt extension, and it can contain no HTML. Once you have created the bypass list, place it on an HTTP server so it can be downloaded to the CacheFlow device.

There are two types of bypass lists: the local bypass list, and the central bypass list.

Local Bypass List

The local bypass list is a list you create and maintain on your network. You can use a local bypass list alone, or in conjunction with a central list.

The gateways specified in the bypass list must be on the same subnet as the CacheOS device.

The local bypass list contains a list of IP addresses, subnet masks, and gateways. It can also define the default bypass gateway to be used by both the local bypass list and central bypass list. The gateways specified in the bypass list must be on the same subnet as the CacheFlow device. When you download a bypass list, the list is stored in the device until it is replaced by downloading a new list. A sample local bypass list is shown below:

```

;define the default gateway for the local and central bypass list
BYPASS_GATEWAY 192.168.10.20

;define addresses to bypass
;IP address      subnet      gateway (or use default gateway)
192.168.10.218   255.255.255.255
192.168.10.120   255.255.255.255
172.16.0.0       255.255.255.0       192.168.10.22
    
```

If you do not specify the BYPASS_GATEWAY, and you do not designate the gateway in the address specification, the CacheFlow device will forward the request to the default gateway defined by the network configuration.

Once the bypass list is created, place it on an HTTP server so it can be downloaded to the device. To download the bypass list, use the **load** command described on page 99.

Central Bypass List

The central bypass list is a shared list of addresses that is used by multiple CacheFlow devices. The central list contains addresses to bypass, but does not specify gateways (because the CacheFlow devices will be located on different subnets, using different gateways). The gateway used for matches in the central bypass list is defined using the **BYPASS_GATEWAY** command in the local bypass list. If there is no **BYPASS_GATEWAY** command, the CacheFlow device will use the default gateway defined by the network configuration.

You can create your own central bypass list to manage multiple CacheFlow devices, or you can use the central bypass list maintained by CacheFlow Technical Support at the following URL:

<http://www.cacheflow.com/support/subscriptions/CentralBypassList.txt>

The central bypass list maintained by CacheFlow contains addresses CacheFlow has identified as using client authentication.

Syntax

bypass-list central-path <url>

bypass-list local-path <url>

bypass-list no [central-path | local-path | notify | subscribe]

bypass-list notify

bypass-list poll-now

bypass-list subscribe

central-path Specifies the location of the central bypass list.

local-path Specifies the location of the local bypass list.

no Clears the central bypass list path, local bypass list path, notify option, or subscribe option.

notify Sends an event notification when the central bypass list is updated.

poll-now	If you have notify enabled, poll-now checks to see if a new central list is available and sends a notification message. If you have subscribe enabled, poll-now checks for a new central list and downloads it if available.
subscribe	Automatically installs a new bypass list when the list changes.

Example

```
CacheOS #(config) bypass-list local 192.168.10.3/files/bypasslist.txt<Enter>
ok
CacheOS #(config)
```

caching

Allows you to modify caching parameters.

When the CacheFlow device retrieves an object from the Web and returns it to the client, the object is considered fresh--the Web cache knows it is fresh because it just retrieved the object from the source. The goal of the Web cache is to keep fresh as many of the objects in the cache as possible, so when the objects are requested the CacheOS can deliver them to the client without having to retrieve them from the source.

To keep objects fresh, the Web cache uses a variety of techniques to learn the update patterns of the objects. It then refreshes the objects in the background, before they are requested by a client. You can define how hard the Web cache works to ensure freshness. If you set the desired freshness to 99%, the Web cache will work harder verifying objects are fresh than if you set the desired freshness to 95%. The higher you set the desired freshness, the higher the percentage of objects will be fresh in the cache when they are requested.

The **caching** command allows you to configure how the CacheOS manages the cached objects.

Syntax

caching

When you enter the **caching** command, the interface displays the config caching prompt, where you can enter caching commands:

```
CacheOS #(config) caching<Enter>
CacheOS #(config caching)
```

The caching commands are listed below:

Command	Parameters	Description
always-verify-source		Always verify object freshness with source.
exit		Return to config prompt.
ftp		Allows you to set FTP caching parameters (see the ftp command on page 72 for additional information).
max-cache-size	<megabytes>	Maximum size object to cache.
negative-response	<minutes>	Cache negative responses.
no	always-verify-source	Negate "always-verify-source."
refresh	automatic bandwidth <Kbps> desired-freshness <percent> no automatic	Refresh parameters.

Example

```
CacheOS # (config) caching <Enter>
CacheOS # (config caching) max-cache-size 20 <Enter>
ok
CacheOS # (config caching)
```

direct-deny-list

Configures the direct or deny settings for forwarding.

When using forwarding, the CacheOS forwards requests for objects not found in the cache to the forwarding gateway. The forwarding gateway then determines what to do with the request.

Direct addresses are addresses the CacheOS should send directly on the network rather than to the forwarding gateway. Deny addresses are addresses to which the CacheOS should deny access. The direct and deny address specifications are made up of a subnet and mask. Requested addresses are compared to the subnet and mask to determine a match. If the request does not match an address in the direct or deny list, the CacheOS sends the request to the gateway.

The direct and deny list is a simple text file containing a list of IP addresses, subnet masks, and commands. A sample direct or deny list is illustrated below:

206.94.0.0	255.255.0.0	DENY
206.96.0.0	255.255.0.0	DENY
206.65.0.0	255.255.0.0	DIRECT

To enter a direct or deny list, create a text file with the direct or deny commands, then place the file on an HTTP server. To download the list to the CacheOS, use the **load** command described on page 99.

Syntax

direct-deny-list path <url>|no path

path <url> The URL from which the direct or deny list can be downloaded.

no path Clears the direct-deny-list URL.

Example

```
CacheOS #(config) direct-deny http://192.168.10.3/files/dodlist.txt<Enter>
ok
CacheOS #(config)
```

dns

Allows you to modify the DNS settings for the Web cache. Note that the alternate DNS servers are only checked if the servers in the standard DNS list return: "Name not found."

Syntax

```
dns alternate <ip>
dns clear [imputing | server]
dns imputing <name>
dns server <ip>
dns no [imputing <imputed name> | server <ip address>]
```

alternate	Add entry to alternate DNS server list
clear	Removes all entries from the imputed names or DNS servers list.
imputing	Adds an entry to the name imputing list.
server	Adds an entry to the DNS server list.
no	Removes an entry from the imputed names or DNS servers list.

Example

```
CacheOS #{config} dns server 192.168.10.64<Enter>
ok
CacheOS #{config}
```

event-log

Allows you to configure the event log settings.

event-log

Syntax

When you enter the **event-log** command, the interface displays the config event-log prompt, where you can enter event-log commands:

```
CacheOS # (config) event-log <Enter>
CacheOS # (config event-log)
```

The event-log commands are listed below:

Command	Parameters	Description
exit		Return to config prompt
level	severe resource informational verbose	Event log level
log-size	<megabytes>	Specify event log size
mail	add <email address> cacheflow-notify clear no [cacheflow-notify smtp-gateway] remove <email address> smtp-gateway <address>	Send email when specific events occur
when-full	overwrite stop	What to do when the event log reaches max size

You must replace the default CacheFlow SMTP gateway with your gateway. If you do not have access to an SMTP gateway, you can use the CacheFlow gateway to send event messages to CacheFlow (the CacheFlow SMTP gateway will only send mail to CacheFlow, it will not forward mail to other domains).

Event-log commands for Syslog

Syslog is an event monitoring scheme that is especially popular in UNIX environments. In sites where Syslog is used, there is typically a “log host”

node, which acts as a “sink” for several devices on the network. You must have a Syslog daemon operating on your network.

Syslog format: “Date Time DNS Name Event”

The event-log commands for Syslog are listed below:

Command	Parameters	Description
disable		Disable Syslog notification
enable		Enable Syslog notification
loghost	<IP> or <name>	Configure Syslog loghost for notifications
no	<loghost>	Negative Syslog commands

Example

```
CacheOS # (config) event-log <Enter>
CacheOS # (config event-log) syslog loghost 206.94.0.0 <Enter>
CacheOS # (config event-log) syslog enable <Enter>
CacheOS # (config event-log) syslog disable <Enter>
CacheOS # (config event-log) syslog no loghost <Enter>
```

Example

```
CacheOS # (config) event-log <Enter>
CacheOS # (config event-log) level informational <Enter>
ok
```

filter-list

Sets the filter list options.

The CacheOS can filter requests made by clients using a filter list. When a filter list is loaded, all requested URLs are compared to the list, and processed based on the results.

The filter list can be used to assign the following actions for a URL:

- deny service
- access direct
- do not refresh
- do not cache
- cache advertising objects
- case-insensitive matching

To use the filter list, create a text file that contains a filter statement for each URL you want to filter. The syntax of the filter statement is shown below:

```
url filter = value
```

The URL can be specified as a full URL, or a regular expression. For information on using regular expressions, see "Regular Expressions" on page 173.

The URL can be a host name or directory path. Some sample URLs are shown below:

```
www.yahoo.com  
www.yahoo.com/quotes/  
www.yahoo.com/news
```

If you specify a directory path, the CacheOS will treat the specification as a full directory name. For example, the URL `www.yahoo.com/news` will not match the URL `www.yahoo.com/newspaper`.

The filters are described below:

Parameters:	Value	Description
service	no	Deny service to the URL.
direct	yes	Do not forward requests to a parent proxy or SOCKS server. This filter only applies when the device is configured to forward requests.
refresh	no	Do not refresh the object if it is cached.
cache	no	Do not cache the object.
advertisement	yes	Cache objects at this URL, and request the objects in the background to maintain the hit count.
case_insensitive	yes	Match URLs using case-insensitivity. By default all URLs are matched in a case-sensitive manner. This filter should be set to match URLs served by Operating Systems such as Windows NT, which is case insensitive.

You can specify multiple filters in a single filter statement. Sample filter statements are shown below:

```
ftp://.* direct=yes
http://www.dilbert.com service=no
http://www.nytimes.com cache=no refresh=no
http://www.engservices.com direct=yes
http://.*.cacheflow.* direct=yes
http://www.msnbc.com case_insensitive=yes cache=no
```

Type each statement on a separate line. You can use the ";" character at the beginning of a line to enter comments. The maximum length of a line is 4096 bytes.

Using the Filter List to Restrict Access

You can also use the filter list to restrict user access to the CacheOS. You can create an ACL (access control list) that defines the IP addresses and subnets that have access to the CacheOS, then eliminate service to all addresses not in the list. The commands required to create an ACL are shown below:

```
# create an ACL named myusers
define acl myusers
    192.168.24.0/24
end acl myusers

# eliminate service to all users not in the myusers ACL
ALL acl=!myusers service=no cache=no
```

This example creates an ACL named “myusers” that contains all addresses on the subnet of 192.168.24.0. It then sets the service and cache options to “no” for anyone not in the list (the exclamation point “!” negates the list). Note that the syntax is case sensitive, and the ALL rule must be in uppercase.

Creating Filter Lists

There are two types of filter lists: the local filter list, and the central filter list.

Local Filter List

The local filter list is a list you create and maintain on your network. You can use a local filter list alone, or in conjunction with a central list.

Central Filter List

The central filter list is a shared list of addresses that is used by multiple CacheFlow devices. You can create your own central filter list to manage multiple CacheFlow devices, or you can use the central filter list maintained by CacheFlow Technical Support at the following URL:

<http://www.cacheflow.com/support/subscriptions/CentralFilterList.txt>

Syntax

```
filter-list central-path <url>
filter-list local-path <url>
filter-list no [central-path | local-path | notify | subscribe]
filter-list notify
filter-list poll-now
filter-list subscribe
```

central-path Specifies the location of the central filter list.

local-path	Specifies the location of the local filter list.
no	Clears the central filter list path, local filter list path, notify option, or subscribe option.
notify	Sends an event notification when the central filter list is updated.
poll-now	If you have notify enabled, poll-now checks to see if a new central list is available and sends a notification message. If you have subscribe enabled, poll-now checks for a new central list and downloads it if available.
subscribe	Automatically installs a new filter list when the list changes.

Once the filter list is created, place it on an HTTP server so it can be downloaded. To download the filter list to the CacheFlow Web cache, use the **load** command described on page 99.

Example

```
CacheOS #(config) filter-list local-path 192.168.10.3/files/filterlist.txt<Enter>
ok
CacheOS #(config)
```

forwarding

Forwards requests to another cache.

The CacheOS can be configured to forward requests to another Web cache or proxy. If a gateway is specified, when an object is requested that is not in the cache, the Web cache will forward the request to the gateway rather than retrieve the object from the network. A primary and alternate gateway can be specified. For detailed information on forwarding, see "Configuring Hierarchical Caches" on page 17.

Syntax

forwarding alternate-gateway

forwarding no [alternate-gateway | primary-gateway]

forwarding primary-gateway

alternate-gateway Configure alternate forwarding gateway settings.

no Remove primary or alternate gateway.

primary-gateway Configure primary forwarding gateway settings.

Example

```
CacheOS #(config) forwarding primary-gateway<Enter>
CacheOS #(config forwarding primary) address 192.168.10.14<Enter>
ok
CacheOS #(config forwarding primary)
```

ftp

Allows you to modify FTP caching parameters. The **ftp** command is available from the caching prompt. To enter the **ftp** command, first enter the **caching** command.

Syntax

ftp

When you enter the **ftp** command, the interface displays the config caching ftp prompt, where you can enter FTP caching commands:

```
CacheOS #(config) caching<Enter>
CacheOS #(config caching) ftp<Enter>
CacheOS #(config caching ftp)
```

The FTP caching commands are listed below:

Command	Parameters	Description
disable		Disable FTP caching.
enable		Enable FTP caching
exit		Return to config caching prompt
max-cache-size	<megabytes>	Maximum size object to cache
type-m-percent	<percent>	The length of time to cache the object based on the last modified date. Expressed as the percentage of time since the object was last modified.
type-n-initial	<hours>	The number of hours to cache objects that do not have an expiration time.

Example

```
CacheOS #(config) caching<Enter>
CacheOS #(config caching) ftp<Enter>
CacheOS #(config caching ftp) max-cache-size 20<Enter>
```

hostname

Sets the Web cache hostname.

You can assign a name to the Web cache. If you have entered a host name for the Web cache in your DNS server, you can use the same name. To set the name, follow the steps outlined below:

Syntax

hostname <name>

<name> The name to use for this Web cache.

Example

```
CacheOS #(config) hostname CacheFlow Demo<Enter>
ok
CacheOS #(config)
```

http-proxy-port

Sets the proxy port for HTTP requests.

The proxy port is the port on which the Web cache listens for HTTP requests. The default port is 8080.

Syntax

http-proxy-port <port>

<port>

The port to use for HTTP requests. The default port is 8080.

Example

```
CacheOS #(config) http-proxy-port 8084<Enter>
ok
CacheOS #(config)
```

icp

Sets the ICP configuration options. For information on creating an ICP or advanced forwarding configuration, see "Configuring Hierarchical Caches" on page 17.

Once you have created the ICP configuration file, place the file on an HTTP server so it can be downloaded to the CacheFlow Web cache. To download the ICP configuration to the Web cache, use the **load** command described on page 99.

Syntax

icp [path <url> | no path]

path <url> The URL from which the configuration file can be downloaded.

no path Clears the ICP configuration path.

Example

```
CacheOS #(config) icp path 192.168.10.3/files/icpconfig.txt<Enter>
ok
CacheOS #(config)
```

interface

Allows you to configure the network interfaces.

The built-in Ethernet adapter is configured for the first time using the setup console. If you want to modify the built-in adapter configuration, or you have multiple adapters, you can configure each one using the command line interface.

Syntax

interface #

The network interface to configure, from 0 - 3.

When you enter the **interface** command, the command-line interface displays the config interface prompt, where you can enter interface configuration commands:

```
CacheOS #(config) interface 0<Enter>
CacheOS #(config interface 0)
```

The interface configuration commands are listed below:

Command	Parameters	Description
accept-inbound		Allow inbound connections on this interface
full-duplex		Configure interface for full duplex
exit		Return to config prompt
half-duplex		Configure interface for half duplex
ip-address	<address>	Set IP address for the interface
instructions	Proxy default-pac central-pac	Configure client proxy instructions
link-autosense		Interface should autosense speed and duplex

Command	Parameters	Description
no	accept-inbound link-autosense	Negative command variations
speed	10 100	Configure speed for interface
subnet-mask	<mask>	Set subnet mask for interface

Example

```
CacheOS # (config) Interface 1 <Enter>
CacheOS # (config interface 1) ip-address 192.168.10.18 <Enter>
ok
CacheOS # (config interface 1)
```

ip-default-gateway

Sets the default IP gateway.

Syntax

ip-default-gateway <address>

<address> The IP address of the default gateway to be used by the Web cache.

Example

```
CacheOS #(config) ip-default-gateway 192.168.10.72<Enter>
ok
CacheOS #(config)
```

line-vty

Allows you to configure the terminal settings for the command line interface.

Syntax

line-vty

When you enter the **line-vty** command, the command-line interface displays the config line vty prompt, where you can enter configuration commands:

```
CacheOS #(config) line-vty<Enter>
CacheOS #(config line-vty)
```

The **line-vty** configuration commands are listed below:

Command	Parameters	Description
exit		Return to the config prompt
length	<number of lines>	Set number of lines on a screen
telnet	no transparent	Telnet protocol specific configuration
timeout	<minutes>	Configure line timeout in minutes

Example

```
CacheOS #(config) line-vty<Enter>
CacheOS #(config line vty) length 60<Enter>
ok
CacheOS #(config line-vty)
```

management-port

Sets the IP port to which the Web cache listens for Web console connections.

Syntax

management-port <port>

<port> The port to use for HTTP requests. The default port is 8081.

Example

```
CacheOS #(config) management-port 8086<Enter>
ok
CacheOS #(config)
```

no

Clears the specified parameter.

Syntax

no ip-default-gateway

no socks-machine-id

no upgrade-path

ip-default-gateway Set the default IP gateway to zero.

socks-machine-id Remove the SOCKS machine id.

upgrade-path Clear the upgrade image download path.

Example

```
CacheOS #(config) no socks-machine-id<Enter>
ok
CacheOS #(config)
```

ntp

Sets NTP parameters.

The CacheOS sets UTC time by connecting to an NTP server. The CacheOS includes a list of NTP servers available on the Internet. If an NTP server is not available, you can set the time manually using the Web interface.

Syntax

ntp clear
ntp enable
ntp disable
ntp server <address>
ntp no server <address>

clear	Remove all entries from NTP server list.
enable	Enable NTP.
disable	Disable NTP.
server	Add entry to NTP server list.
no	Remove entry from NTP server list.

Example

```
CacheOS #(config) ntp server clock.tricity.wsu.edu<Enter>
ok
CacheOS #(config)
```

restart

Sets restart options for the Web cache. To restart the Web cache, exit config mode and enter the **restart regular** command.

Syntax

```
restart compress
restart core-image [context | full | none]
restart mode [hardware | software]
restart no compress

compress          Specify compressed core image.
core-image        Specify type of core image to write.
mode              Configure hard or soft restart.
no                Negative restart commands.
```

Example

```
CacheOS #(config) restart mode software<Enter>
ok
CacheOS #(config)
```

return to sender

Enables the return to sender option for inbound and/or outbound connections. We recommend that you consult with CacheFlow Customer Support before you attempt to use the return-to-sender option.

When enabled for a connection, CacheOS records the previous hop MAC address and associates it with the source IP. This allows the cache to use the previously stored hop when sending packets if the next hop from the routing table is the default gateway. There is no distinction made on port or protocol.

Syntax

return-to-sender inbound enable

return-to-sender inbound disable

return-to-sender outbound enable

return-to-sender outbound disable

The return-to-sender configuration commands are listed below:

Command	Parameters	Description
inbound enable		Enable "return to sender" for incoming sessions
outbound enable		Enable "return to sender" for outgoing sessions
inbound disable		Disable "return to sender" for incoming sessions
outbound disable		Disable "return to sender" for outgoing sessions

Example

```
CacheOS # (config) return-to-sender inbound enable<Enter>
ok
CacheOS # (config)
```

rip

Sets the RIP configuration options.

The RIP configuration is defined in a configuration file. To configure RIP, first create a text file of RIP commands, then load the file using the **load** command described on page 99. The RIP commands are described on page 156.

Syntax

rip enable
rip disable
rip path <url>
rip no path

enable	Enables the RIP configuration.
disable	Disables the RIP configuration
path <url>	The URL from which the RIP configuration file can be downloaded.
no path	Clears the RIP configuration path.

Example

```
CacheOS # (config) rip path 192.168.10.3/files/rip.txt<Enter>
ok
CacheOS # (config)
```

security

Sets security options for the Web cache.

Note that the CacheFlow device can limit proxy services to only those users with proper credentials. See the Technical Note available on the CacheFlow website for details that describe how to create and upload a password file. Once the password file is loaded into the CacheFlow device, you can enable Client Authentication.

Syntax

security allowed-access <address> [mask]

security client-authentication <enable/disable>

security enforce-console-acl

security enable-password

security no

security password <password>

security user-name <user name>

allowed-access	Add the IP address and optional subnet mask to console access control list.
client authentication	Enable/disable proxy user authentication.
enforce-console-acl	Enforce console access control list.
enable-password	Specify console enable password.
no	Remove security setting.
password	Specify console account password.
user-name	Specify console account username.

Example

```
CacheOS #(config) security enable-password syssecure<Enter>
ok
CacheOS #(config)
```

snmp

Sets SNMP options for the Web cache.

The CacheFlow Web cache can be viewed using an SNMP management station. The CacheOS supports MIB-2 (RFC 1213).

Syntax

snmp

The snmp configuration commands are listed below:

Command	Parameters	Description
authorize-traps		Enable SNMP authorize traps.
disable		Disable SNMP.
enable		Enable SNMP.
no		Clear certain SNMP parameters.
reset		Reset SNMP configuration.
read-community	<password>	Specify read community string.
sys-contact	<string>	Set "sysContact" MIB variable.
sys-location	<string>	Set "sysLocation" MIB variable.
trap-address	[1 2 3] <address>	Specify IP address to receive traps.
trap-community	<password>	Specify trap community string.
write-community	<password>	Specify write community string.

Example

```
CacheOS #(config) snmp<Enter>  
CacheOS #(config snmp) trap-address 1 192.168.10.112  
ok  
CacheOS #(config)
```

socks-machine-id

Sets the machine id for SOCKS.

If you are using a SOCKS server for the primary or alternate gateway, you must specify the CacheFlow Web cache's machine ID for the Identification (Ident) protocol used by the SOCKS gateway.

Syntax

socks-machine-id <machine id>

<machine id> The SOCKS machine ID.

static-routes

Sets the network path to download the static routes configuration file.

The CacheFlow device can be configured to use static routes. To use static routes you must create a routing table and place it on an HTTP server accessible to the CacheFlow device. The routing table is a text file that contains a list of IP addresses, subnet masks, and gateways. When you download a routing table, the table is stored in the device until it is replaced by downloading a new table.

The routing table is a simple text file containing a list of IP addresses, subnet masks, and gateways. A sample routing table is illustrated below:

206.63.0.0	255.255.0.0	206.63.158.213
206.64.0.0	255.255.0.0	206.63.158.213
206.65.0.0	255.255.0.0	206.63.158.226

When a routing table is loaded, all requested addresses are compared to the list, and routed based on the best match.

Once the routing table is created, place it on an HTTP server so it can be downloaded to the device. To download the routing table to the CacheOS, use the **load** command described on page 99.

Syntax

static-routes path <url>

static-routes no path

path <url>

The URL from which the static routes configuration file can be downloaded.

no path

Clears the static-routes configuration path.

Example

```
CacheOS #(config) static-routes path 192.168.10.3/files/routes.txt<Enter>
ok
CacheOS #(config)
```

upgrade-path

Sets the network path to the Web cache upgrade. To upgrade the Web cache, use the **load** command described on page 99 then use the **restart upgrade** command to start the new upgrade software.

Syntax

upgrade-path <url>

<url> The URL where the upgrade is located.

Example

```
CacheOS #(config) upgrade-path http://192.168.10.3/download.htm<Enter>
ok
CacheOS #(config)
```

WCCP

The CacheFlow device can be configured to participate in a WCCP (Web Cache Control Protocol) scheme, where a WCCP-capable router collaborates with a set of WCCP-configured CacheFlow devices to service requests. WCCP is a Cisco-developed protocol. For more information about WCCP, refer to the *Appendix B: WCCP (Web Cache Control Protocol)*.

Once you have created the WCCP configuration file, place the file on an HTTP server so it can be downloaded to the CacheFlow Web cache. To download the WCCP configuration to the Web cache, use the **load** command described on page 99.

Syntax

wccp enable
wccp disable
wccp path <url>
wccp no path

enable	Enables the WCCP configuration.
disable	Disables the WCCP configuration
path <url>	The URL from which the WCCP configuration file can be downloaded.
no path	Clears the WCCP configuration path.

Example

```
CacheOS #{config} wccp path 192.168.10.3/files/wccp.txt<Enter>  
ok  
CacheOS #{config}
```

web-management

Enables and disables the Web interface management console. When **web-management** is disabled, you can still access the CacheOS homepage and online documentation. Only the management and statistics applications are disabled.

Syntax

web-management [enable | disable]

enable Enables the Web interface management console.

disable Disables the Web interface managemnt console.

Example

```
CacheOS # (config) web-management disable<Enter>
ok
CacheOS # (config)
```

Management Commands

The management commands allow you to test and manage the Web cache configuration. The management commands are listed below:

Command	Description
acquire-utc	Acquire UTC from NTP server
clear-cache	Clear the contents of the cache
display	Display a text based url
kill	Terminates a CLI session
load	Load an installable list
offline-disk	Take a disk offline
purge-dns-cache	Clear the DNS cache
ping	Send echo messages
restart	Restart system
restore-defaults	Restore to default configuration
test	Tests the CacheOS subsystems
tracroute	Trace route to destination
upload-access-log	Upload the access log to configured host

acquire-utc

Sets UTC (universal time coordinates) time from the configured NTP server.

Syntax

acquire-utc

The **acquire-utc** command has no keywords or parameters.

Example

```
CacheOS # acquire-utc<Enter>  
ok  
CacheOS #
```

clear-cache

Clears the contents of the Web cache.

Syntax

clear-cache

The **clear-cache** command has no keywords or parameters.

Example

```
CacheOS # clear-cache<Enter>
ok
CacheOS #
```

display

Displays a text-based URL. This command can be used to view the contents of configuration file before it is loaded.

Syntax

display <url>

<url> The URL to display.

Example

```
CacheOS # display http://192.168.10.3/files/dodlist.txt<Enter>
206.94.0.0 255.255.0.0 DENY
206.96.0.0 255.255.0.0 DENY
206.65.0.0 255.255.0.0 DIRECT
CacheOS #
```

kill

Terminates a command line session. You can use the **show sessions** command to display a list of active sessions.

Syntax

kill session_id

session_id The session to terminate.

Example

CacheOS # **show sessions**<Enter>

Sessions:

#	state	type	start	elapsed
01*	PRIVL	telnet	03 May 1999 20:39:04 UTC	00:06:59
02	PRIVL	telnet	03 May 1999 20:29:22 UTC	00:16:18
03	IDLE			
04	IDLE			
05	IDLE			

CacheOS # **kill 2**<Enter>

ok

CacheOS #

load

Load an installable list.

Syntax

load bypass-list central | local

load direct-deny-list

load filter-list central | local

load icp-settings

load rip-settings

load static-route-table

load upgrade

bypass-list Download a new local or central bypass list.

direct-deny-list Download new DCS settings.

filter-list Download a new local or central filter list.

icp-settings Download new ICP settings.

rip-settings Download new RIP settings.

static-route-table Download new static route table.

upgrade Download new system image.

Example

```
CacheOS # load direct-deny-list<Enter>
ok
CacheOS #
```

offline-disk

Takes the specified disk offline. You can use the **show disk** command to display disk status information.

Syntax

offline-disk #

The disk to take offline.

Example

```
CacheOS # offline-disk 3<Enter>
ok
CacheOS #
```

purge-dns-cache

Clears all entries from the DNS cache.

Syntax

purge-dns-cache

The **purge-dns-cache** command has no keywords or parameters.

Example

```
CacheOS # purge-dns-cache<Enter>  
ok  
CacheOS #
```

ping

Pings the specified host.

Syntax

ping <address> | <hostname>

<address> The IP address to ping.

<hostname> The IP host to ping.

Example

CacheOS # **ping 192.168.24.112**<Enter>

Type escape sequence to abort.

Sending 5, 100-byte ICMP Echos to 192.168.10.112, timeout is 2 seconds:

!!!!

Success rate is 100 percent (5/5), round-trip min/avg/max = 67/77/101 ms

Number of duplicate packets received = 0

CacheOS #

restart

Restarts the Web cache.

Syntax

restart [regular | upgrade]

regular Reboot the system according to the restart settings.

upgrade Reboot the system to start running a new image after upgrading.

Example

```
CacheOS # restart regular<Enter>
ok
CacheOS #
```

restore-defaults

Restores the default configuration.

Syntax

restore-defaults

The **restore-defaults** command has no keywords or parameters.

Example

```
CacheOS # restore-defaults<Enter>
ok
CacheOS #
```

test

Tests the CacheOS subsystems.

Syntax

test http [get <url> | loopback]

http get <url>	Starts an internal client to request the specified URL through the CacheOS. The results include the HTTP response and measured throughput.
http loopback	Starts an internal client and HTTP server to perform a loopback test through the CacheOS. If this test fails, there is a hardware or software problem with your device, and you should contact CacheFlow Technical Support for assistance.

Example

```
CacheOS # test http loopback<Enter>
Type escape sequence to abort.
Executing HTTP loopback test
Measured throughput rate is 148.98 Kbytes/sec
HTTP loopback test passed
CacheOS #
```

traceroute

Performs a network trace to the specified destination.

Syntax

traceroute <address> | <hostname>

<address> The IP address to trace.

<hostname> The IP host to ping.

Example

```
CacheOS # traceroute 192.168.24.112<Enter>
Type escape sequence to abort.
Tracing the route to 192.168.24.112
 1 192.168.10.72 4 2 2
 2 192.168.24.15 5 5 5
CacheOS #
```

upload-access-log

Uploads the current access log to the access log upload site.

Syntax

upload-access-log

The **upload-access-log** command has no keywords or parameters.

Example

```
CacheOS # upload-access-log<Enter>  
ok  
CacheOS #
```

Chapter 6

Using the Web Interface

The Web interface is a set of Java-based applets that allow you to configure and manage the CacheOS from a Web browser. This chapter describes how to use the Web interface.

Overview

The Web interface is a set of Web pages and Java applets stored on the Web cache. From the Web interface, you can monitor and configure the device.

The Web interface contains both management and statistics applets, along with documentation. The Java-based applets are accessible from any Web browser that supports Java and JavaScript, such as Netscape Navigator® 3.0 or later, and Microsoft Internet Explorer® 3.0 or later.

The management tasks are organized into basic management tasks that most users will perform, advanced tasks, and statistics. For additional information on the Web interface, see the Online Help included with each applet.

Basic Management Tasks

The CacheOS includes a variety of settings you can modify to configure the device for your environment. The basic management tasks are listed below:

Basic Tasks	Page
Configuring the network adapters	112
Entering DNS servers	113
Setting IP ports	114
Setting the Web cache name	115
Selecting client instructions	116
Setting the time	117
Setting a console username and password	118
Setting access restrictions	119
Enabling SNMP	120
Enabling the access log	119
Setting the access log upload site	123
Setting the access log schedule	124
Setting the access log format	125
Enabling event notification	126
Restarting the device	127
Restoring system defaults	128
Purging the DNS cache	129
Clearing the system cache	129
Upgrading the system	130
Displaying the event log	131

Advanced Management Tasks

The CacheOS includes a number of advanced settings that allow you to optimize performance in a more complex environment. The advanced management tasks are listed below:

Advanced Tasks	Page
Using name imputing	132
Configuring cache options	134
Setting cache policies	136
Setting FTP caching options	138
Using ICP and Advanced Forwarding	139
Using Simple Forwarding	141
Using a SOCKS server	143
Creating a direct or deny list	145
Using a filter list	146
Defining static routes	151
Using a bypass list	153
Using RIP	156

Statistics

The Web interface includes a variety of status and statistics reporting applets. The most common tasks are listed below:

Statistics	Page
Displaying the system status	156
Displaying the number of objects and bytes served	163
Displaying active client connections	164
Displaying how resources are currently being used	165
Displaying cache efficiency	166
Displaying the cache contents distribution	167

Configuring the Network Adapters

The built-in Ethernet adapter is configured for the first time using the Serial Console. If you want to modify the built-in adapter configuration, or you have multiple adapters, you can configure each one using the Web interface. To configure adapter settings, follow the steps outlined below:

1. Select Management from the CacheOS home page.
2. Select the adapter from the adapter drop-down list in the Network applet.
3. Enter the IP address and subnet mask for the adapter.
4. To configure link settings or restrict inbound connections for the adapter, click the Advanced button. Enter your changes and click the Ok button to save the changes.
5. Click the Apply button to save your changes.

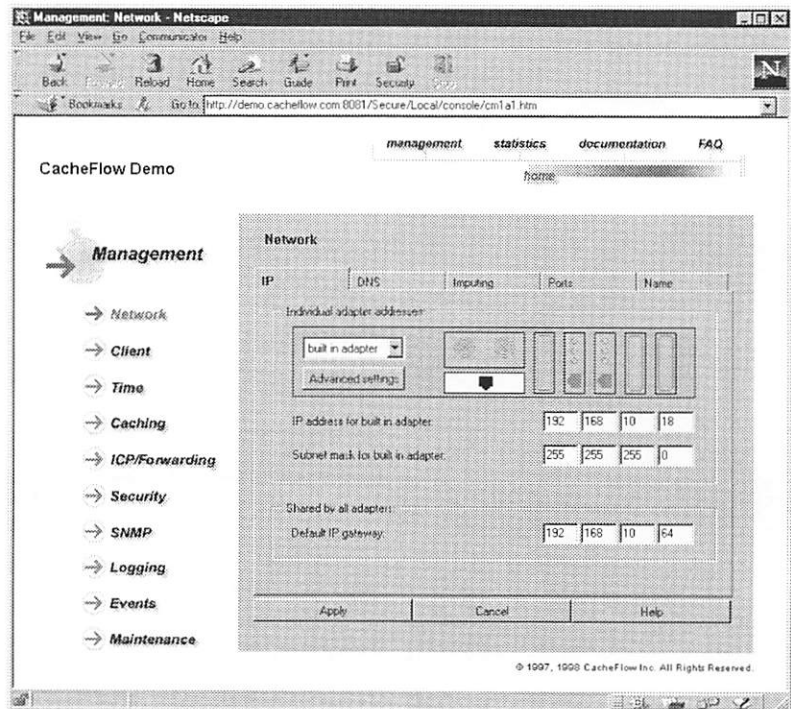


Figure 6-1
Configuring Adapters

Select the adapter to
configure from the adapter
drop-down list.

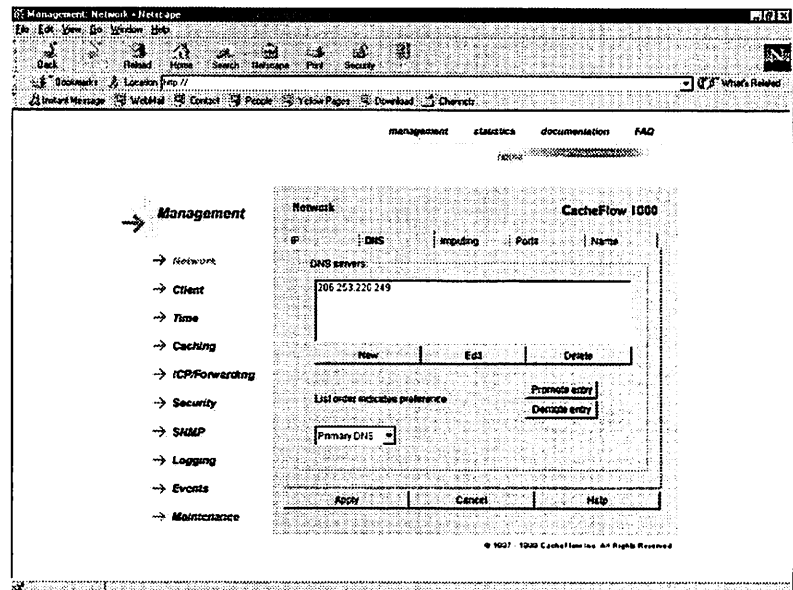
Entering DNS Servers

The primary DNS server is entered using the Serial Console. You can define additional DNS servers using the Web interface. Note that the alternate DNS servers are only checked if the servers in the standard DNS list return: "Name not found."

To enter additional DNS servers, follow the steps outlined below:

1. Select Management from the CacheOS home page.
2. Select the DNS tab in the Network applet.
3. To enter additional DNS servers click the New button.
4. Enter the IP address of the DNS server and click the Ok button.
5. Click the Apply button to save your changes.

Figure 6-2
Entering DNS Servers
Enter additional DNS
servers by clicking
the New button.



Setting IP Ports

You can set the IP ports the CacheOS uses to listen for HTTP requests and the Web interface. The default port for HTTP requests is 8080. The default port for the Web interface is 8081. To change the IP ports, follow the steps outlined below:

1. Select Management from the CacheOS home page.
2. Select the Ports tab in the Network applet.
3. Enter the port to use for HTTP requests.
4. Enter the port to use for the Web interface.
5. Click the Apply button to save your changes.

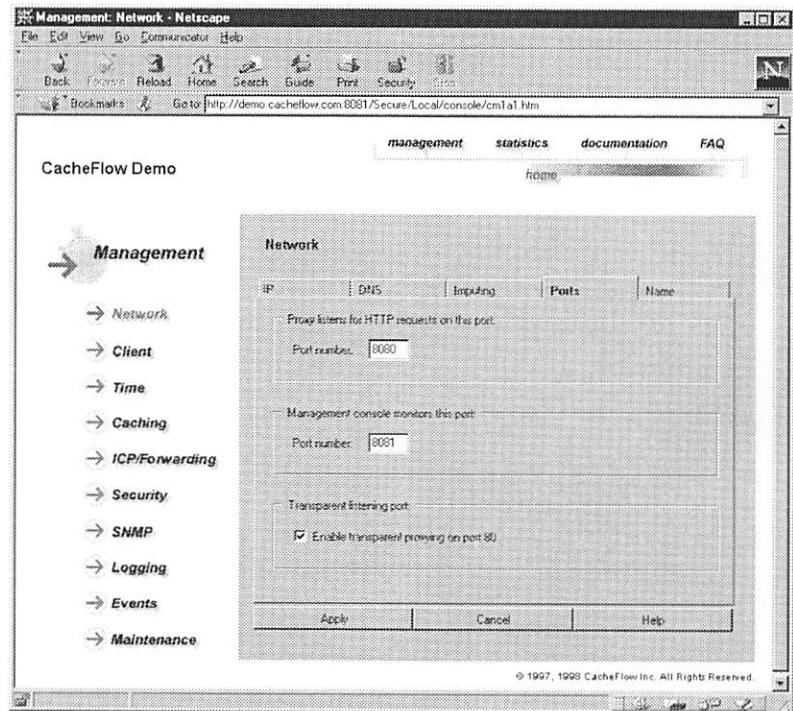


Figure 6-3
Configuring IP Ports

*You can set the port used
for the HTTP requests and
the Web interface.*

Setting the CacheFlow Web Cache Name

You can assign a name to the Web cache. If you have entered a host name for the Web cache in your DNS server, you can use the same name. To set the name, follow the steps outlined below:

1. Select Management from the CacheOS home page.
2. Select the Name tab in the Network applet.
3. Enter the name in the name field.
4. Click the Apply button to save your changes.

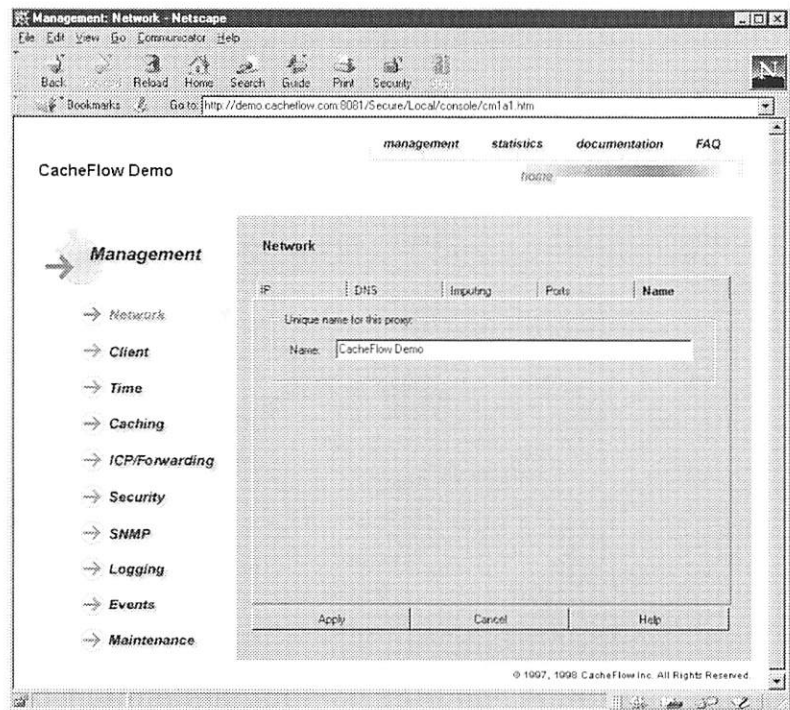


Figure 6-4
Setting the
CacheFlow Web Cache
Name

*Type the name in the
name field.*

Selecting Client Instructions

You can select the client instructions the CacheOS displays when a user selects Browser Configuration from the CacheOS home page. These instructions describe how to configure various browsers to use the CacheFlow Web cache. There are instructions for each network adapter. To select the client instructions, follow the steps outlined below:

1. Select Management from the CacheOS home page.
2. Select the Client applet.
3. Select the adapter you want to configure in the adapter drop-down list.
4. Select the instructions you want to display.
5. Click the Apply button to save your changes.

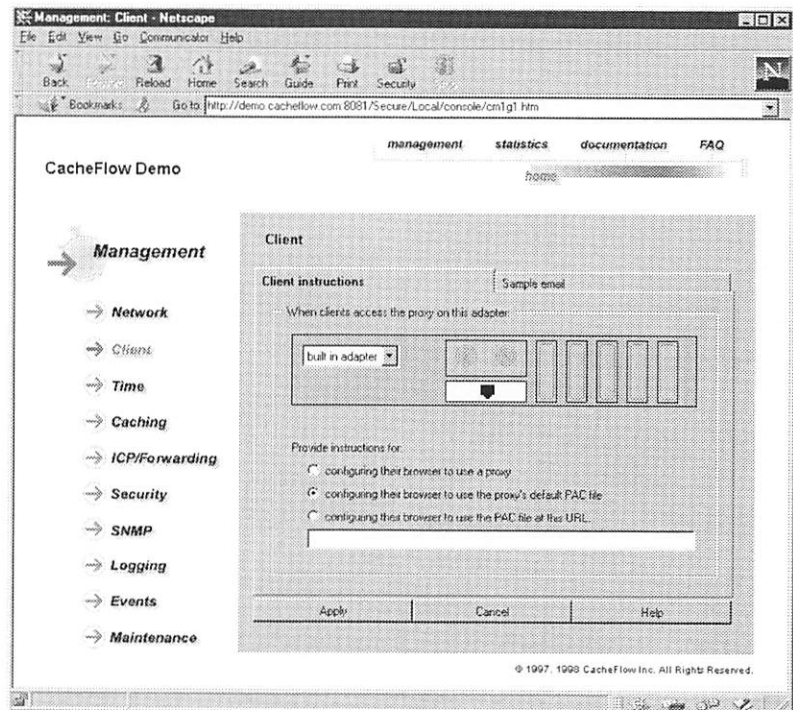


Figure 6-5
Selecting Client Instructions

You can select the instructions to display for configuring the client browser.

Setting the Time

The CacheOS sets UTC time by connecting to an NTP server. The CacheOS includes a list of NTP servers available on the Internet. If an NTP server is not available, you can set the time manually. To set the time manually, follow the steps outlined below:

1. Select Management from the CacheOS home page.
2. Select the Time applet.
3. Clear the Enable NTP checkbox.
4. Click the Pause button.
5. Enter the current UTC time and date in the UTC fields.
6. Click the Resume button.
7. Click the Apply button to save your changes.

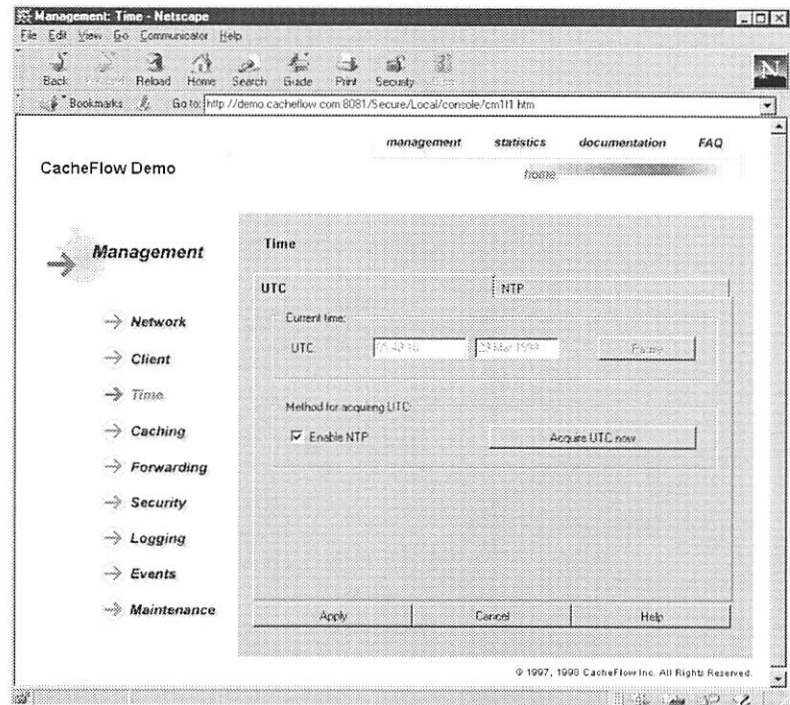


Figure 6-6
Setting the Time
The CacheOS retrieves the UTC time from an NTP server if available.

Setting the Console Username and Password

The command line interface and Web interface can be protected with a username and password. The username and password is set for the first time using the Serial Console. If you forget the password, you can use the Serial Console to reset it. To set the username and password using the Web interface, follow the steps outlined below:

1. Select Management from the CacheOS home page.
2. Select the Security applet.
3. Enter the username in the Username field.
4. Enter the password in the Password field.
5. Re-enter the password in the Verify field.
6. Click the Apply button to save your changes.

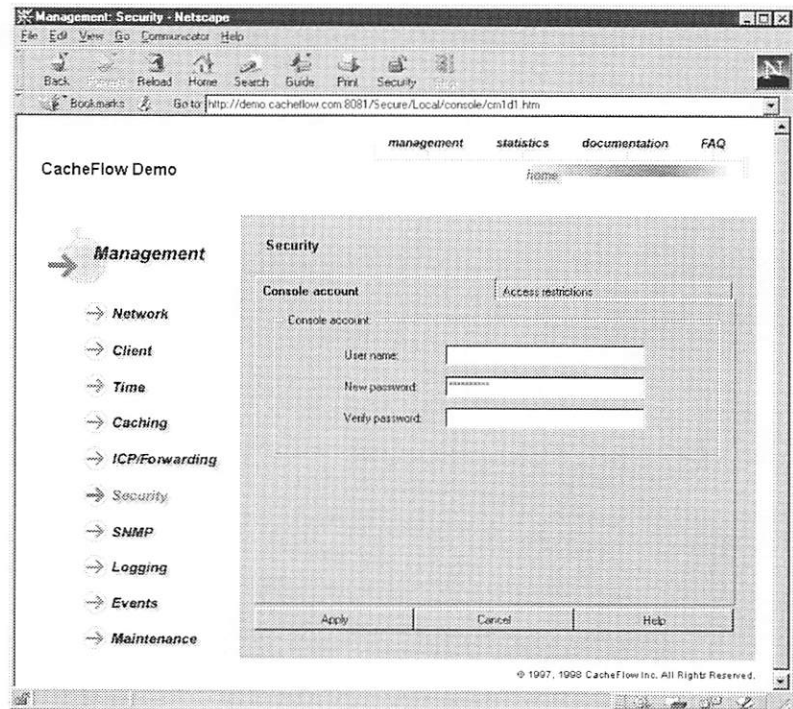


Figure 6-7
Setting the Management
Console Password
The username and
password can be changed
using the Web interface.

Setting Access Restrictions

Important!

Before you enable the access control list, verify the station restrictions you have entered will not restrict your current workstation from accessing the Management Console. If you set the access restrictions incorrectly from the Management Console you can correct the problem using the Serial Console.

The CacheOS allows you to restrict access to the Management Console by station address. You can restrict access to individual IP addresses, or subnets. To create an access control list and restrict access to the Management Console, follow the steps outlined below:

1. Select Management from the CacheOS home page.
2. Select the Security applet.
3. Select the Access restrictions tab.
4. Click the new button then enter the IP address specification in the Subnet field.
5. Enter the subnet mask in the Mask field. To restrict access to an individual workstation, enter 255.255.255.255.
6. Click the Ok button.
7. Enable the Engage console access control list checkbox.
8. Click the Apply button to save your changes.

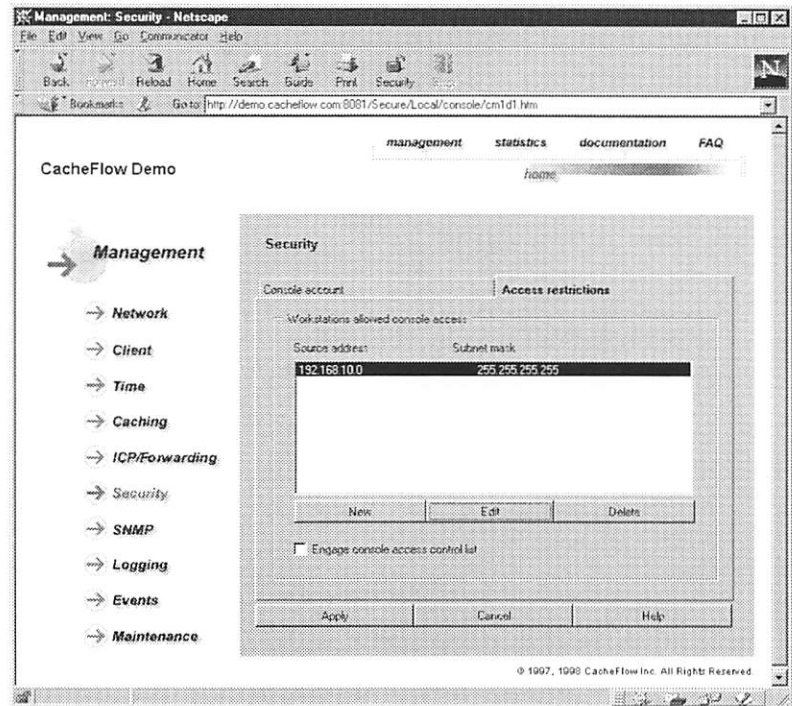


Figure 6-8
Access Control List

The CacheOS allows you to restrict access to individual machines or subnets.

Setting Proxy User Authentication

The CacheFlow device can limit proxy services to only those users with proper credentials.

See the Technical Note available on the CacheFlow website for details that describe how to create and upload a password file. Once the password file is loaded into the CacheFlow device, you can enable Client Authentication.

To enable Client Authentication follow the steps outlined below:

1. Select Management from the CacheOS home page.
2. Select the Security applet.
3. Select the Client authentication tab.
4. Click the Enable client authentication box.
5. If you want to change the default public community strings, go to the Community strings tab and enter the new values.
6. Click the Apply button to save your changes.

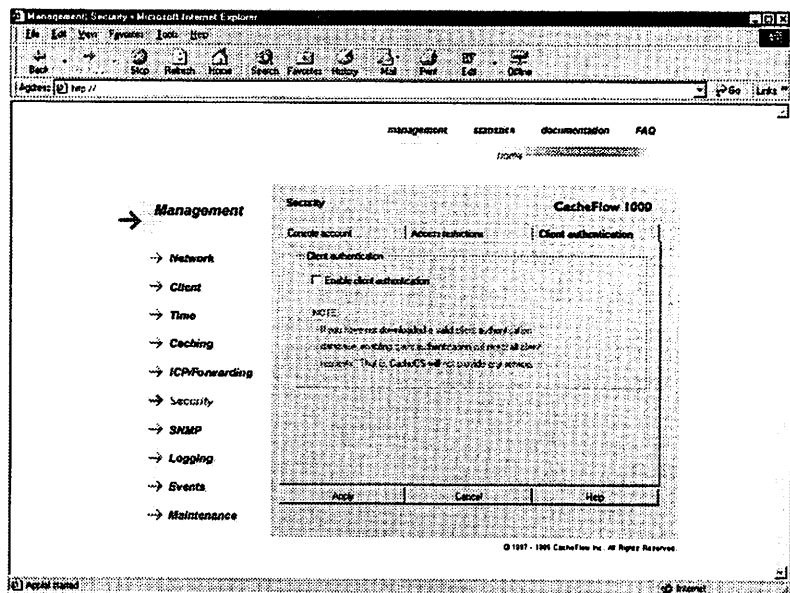


Figure 6-9
Setting Proxy User
Authentication

Enabling SNMP

The CacheFlow Web cache can be viewed using an SNMP management station. The CacheOS supports MIB-2 (RFC 1213). To enable and configure SNMP, follow the steps outlined below:

7. Select Management from the CacheOS home page.
8. Select the SNMP applet.
9. Check the Enable SNMP checkbox.
10. If you want to change the default public community strings, go to the Community strings tab and enter the new values.
11. Click the Apply button to save your changes.

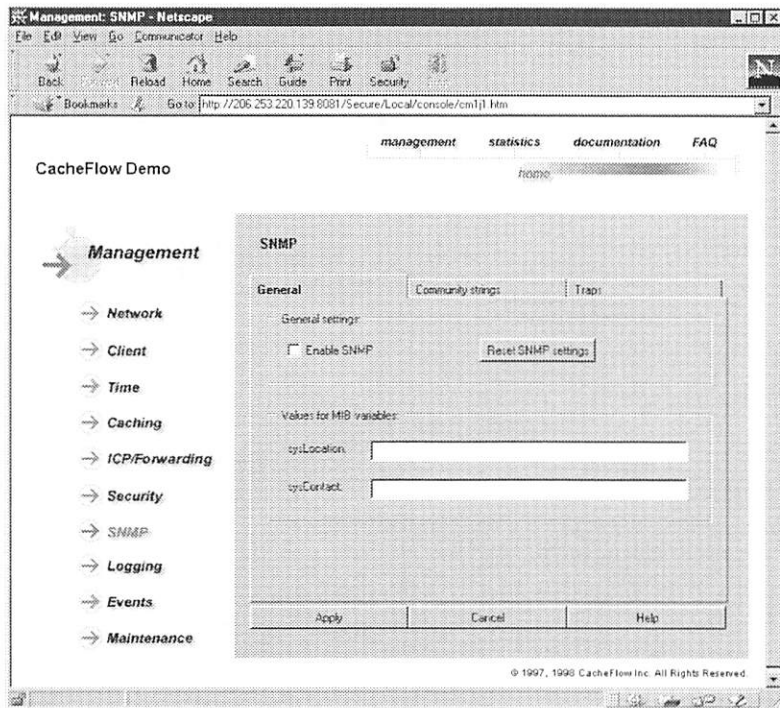


Figure 6-10
Enabling SNMP

When SNMP is enabled you can view the CacheFlow Web cache using a standard SNMP browser.

Enabling the Access Log

The CacheOS can maintain an access log for each HTTP request made. The access log can be stored in one of two formats, which can be read by a variety of reporting utilities (see "Setting the Access Log Format" on page 125 for additional information). To enable the access log, follow the steps outlined below:

1. Select Management from the CacheOS home page.
2. Select the Logging applet.
3. Select the Enable URL access logging checkbox on the General tab.
4. Click the Apply button to save your changes.

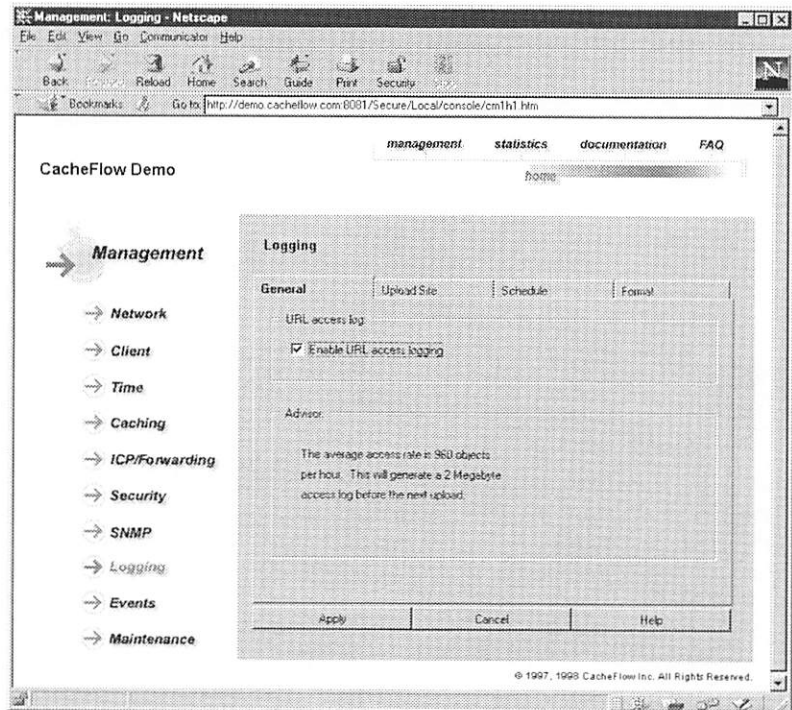


Figure 6-11
Enabling the Access Log
The CacheOS maintains an access log that tracks all HTTP requests.

Setting the Access Log Upload Site

The CacheOS uploads the access log to an FTP server based on the upload schedule. You must specify the FTP server address, directory, and login information. To configure the access log upload site, follow the steps outlined below:

1. Select Management from the CacheOS home page.
2. Select the Logging applet.
3. Select the Upload Site tab.
4. Enter the Filename prefix if needed.
5. Enter the FTP server address in the Host field.
6. Enter the directory path on the FTP server in the Path field.
7. Enter the username to log into the FTP server in the username field.
8. Enter the password for the username in the password field.
9. Click the Apply button to save your changes.

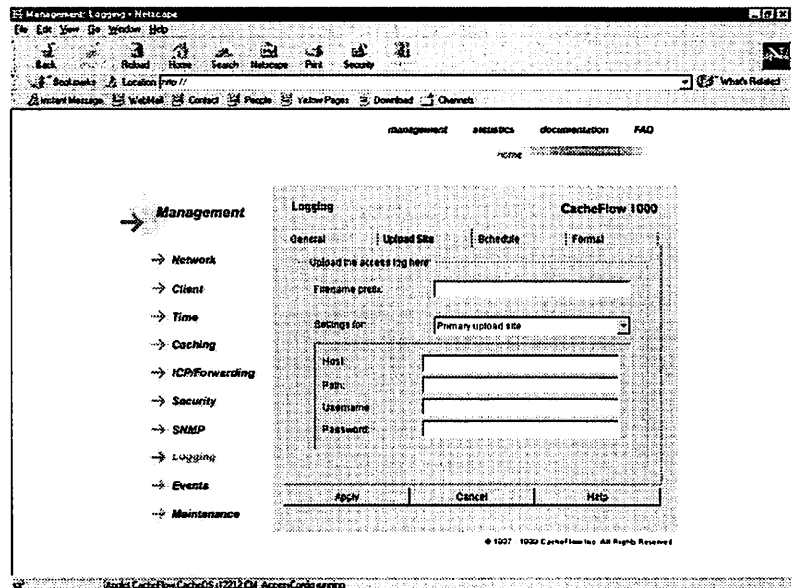


Figure 6-12
Setting the Access Log
Upload Site

The CacheOS uploads the access log to an FTP server.

Setting the Access Log Schedule

The CacheOS uploads the access log to an FTP server based on the upload schedule. To set the upload schedule, follow the steps outlined below:

1. Select Management from the CacheOS home page.
2. Select the Logging applet.
3. Select the Schedule tab.
4. Set the schedule.
5. Set the contingency options.
6. Click the Apply button to save your changes.
7. Select the General tab to view an estimate of how large the log will grow based on the current schedule.

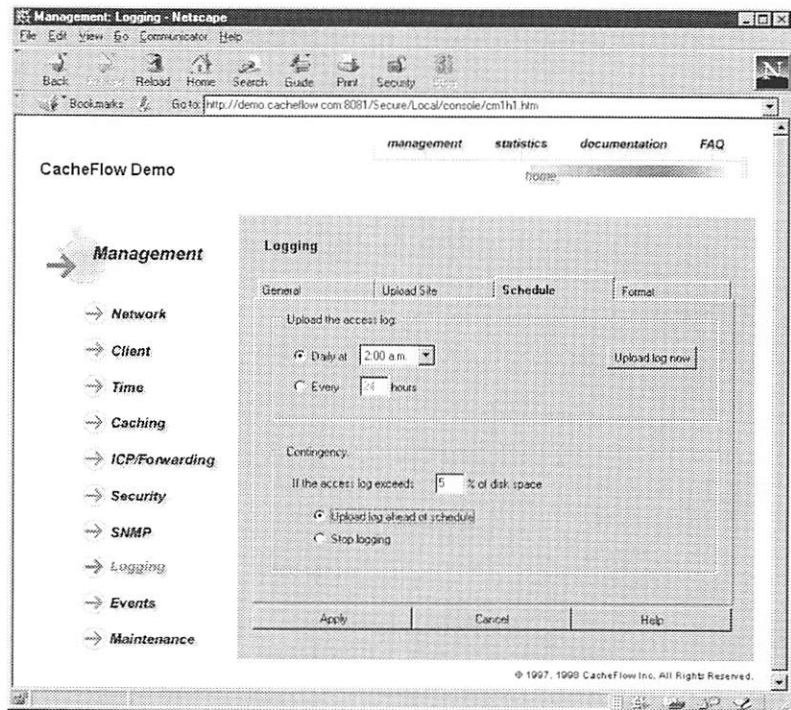


Figure 6-13
Setting the Access Log
Schedule

The CacheOS uploads the access log according to the access log schedule.

Setting the Access Log Format

The CacheOS can save the access log in one of three formats: common log format, the Squid-compatible format, or a custom format. See Appendix A on page 169 for information on the log formats. To set the access log format, follow the steps outlined below:

1. Select Management from the CacheOS home page.
2. Select the Logging applet.
3. Select the Format tab.
4. Select the format you want to use.
5. Click the Apply button to save your changes.
6. Select the General tab to view an estimate of how large the log will grow based on the current schedule.

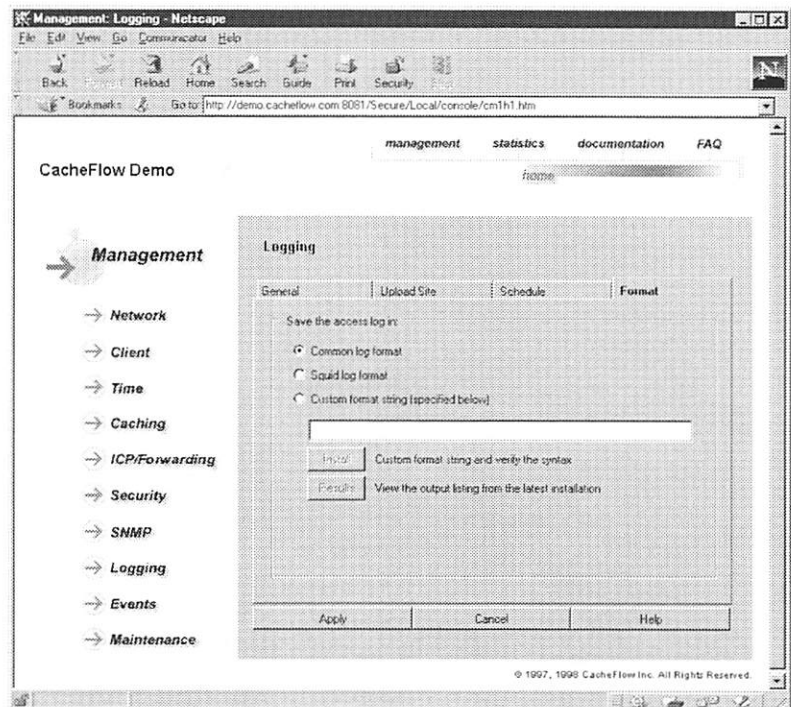


Figure 6-14
Setting the Access Log
Format

The CacheOS can create the access log in the common log format or Squid-compatible format.

Enabling Event Notification

The CacheOS can notify you by email if a device failure occurs. To enable event notifications, follow the steps outlined below:

1. Select Management from the CacheOS home page.
2. Select the Events applet.
3. Select the Mail tab.
4. Click the New button to add a new email address.
5. Enter the host name of your mail server in the SMTP gateway name field, or enter the IP address of your mail server in the SMTP gateway IP field. You must replace the default CacheFlow SMTP gateway with your gateway. If you do not have access to an SMTP gateway, you can use the CacheFlow gateway to send event messages to CacheFlow (the CacheFlow SMTP gateway will only send mail to CacheFlow, it will not forward mail to other domains).
6. Click the Apply button to save your changes.

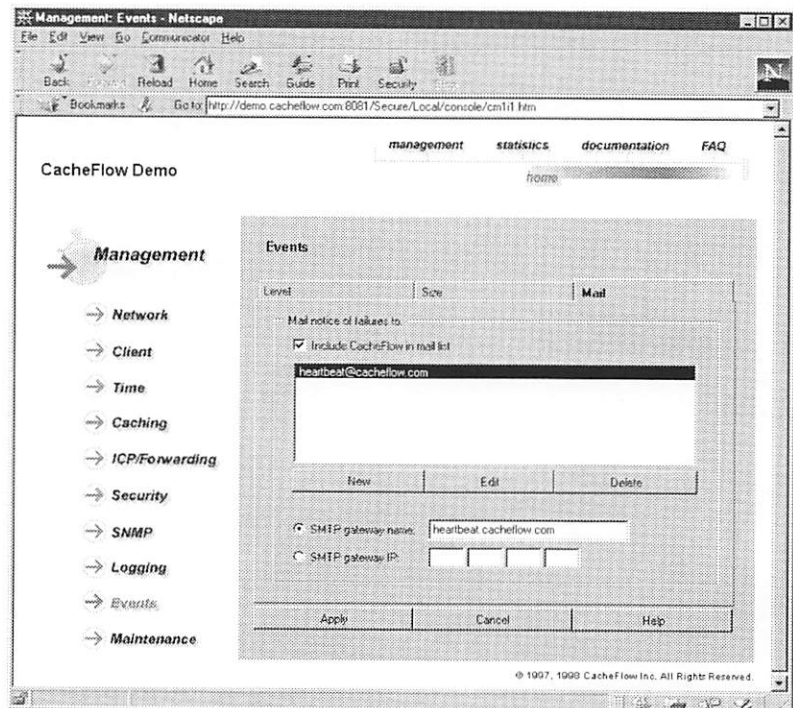


Figure 6-15
Enabling Event
Notifications

*The CacheOS can email
event notifications to a list
of email addresses.*

Restarting the Device

You can restart the CacheOS from a remote location using the Maintenance applet. When you restart the device, you can choose a quick, software-only restart, or a complete hardware and software restart. The hardware and software restart may take several minutes depending on the device configuration. To restart the device, follow the steps outlined below:

1. Select Management from the CacheOS home page.
2. Select the Maintenance applet.
3. Select the Restart tab.
4. Select the type of restart, software only or hardware and software restart.
5. Click the Apply button to save your changes.
6. Click the Restart Now button to restart the device.
7. Click the Ok button to confirm.

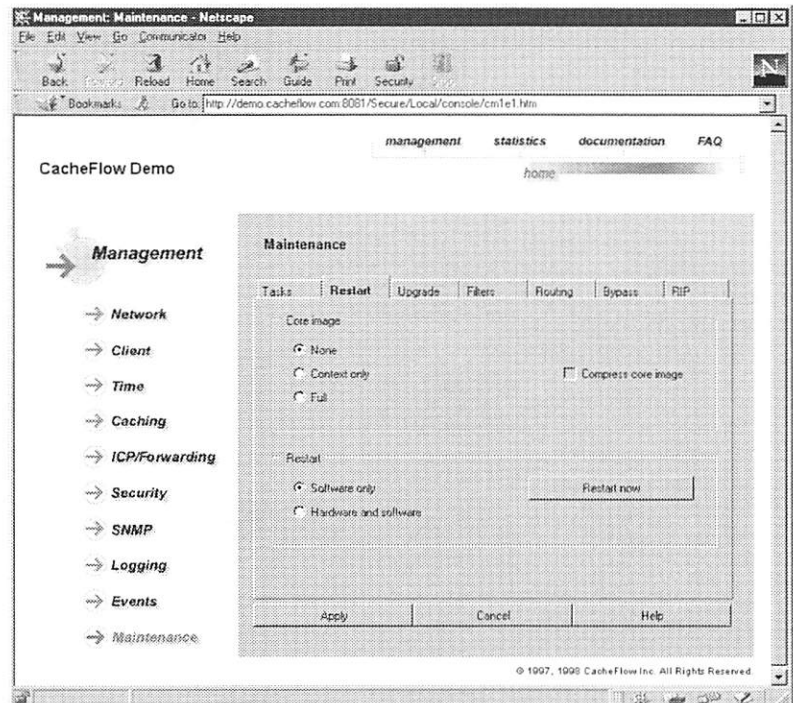


Figure 6-16
Restarting the Device
You can use the Maintenance applet to restart the CacheOS.

Restoring System Defaults

Important!

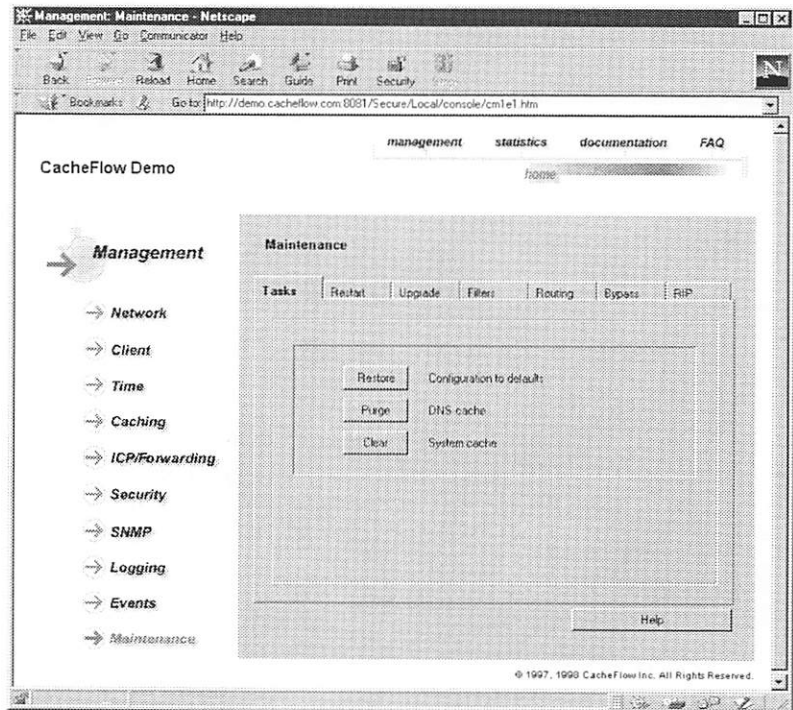
If you restore system defaults you must go to the Network applet to reset the IP address, default gateway, and DNS server addresses before restarting the CacheOS.

You can restore the system configuration to default values using the Maintenance applet. When you restore system defaults, the IP address, default gateway, and the DNS server addresses will be cleared, and you will have to go to the Network applet to set them. To restore defaults, follow the steps outlined below:

1. Select Management from the CacheOS home page.
2. Select the Maintenance applet.
3. Click the Restore button to restore defaults.
4. Click the Ok button to confirm.

Figure 6-17
Restoring Defaults

You can use the Maintenance applet to restore system defaults, purge the DNS cache, and clear the system cache.



Purging the DNS Cache

You can purge the DNS cache maintained by the CacheOS at any time. You may need to purge the DNS cache if you have experienced a problem with your DNS server, or if you have changed your DNS configuration. To purge the DNS cache, follow the steps outlined below:

1. Select Management from the CacheOS home page.
2. Select the Maintenance applet.
3. Click the Purge button to purge the DNS cache.
4. Click the Ok button to confirm.

Clearing the System Cache

You can clear the system cache at any time. When you clear the system cache, all objects in the cache are set to expired. The objects are not immediately removed from memory or disk, but each object that is requested is retrieved from the source before it is served. To purge the system cache, follow the steps outlined below:

1. Select Management from the CacheOS home page.
2. Select the Maintenance applet.
3. Click the Clear button to clear the system cache.
4. Click the Ok button to confirm.

Upgrading the System

You can upgrade the CacheOS system using the Management applet. To upgrade the system, follow the steps outlined below:

1. Select Management from the CacheOS home page.
2. Select the Maintenance applet.
3. Select the Upgrade tab.
4. Enter the URL where the system software is located, or click the Show me button to display a list of available software.
5. Click the Download button to begin the upgrade.
6. Click the Restart button then click OK to start running the new version.

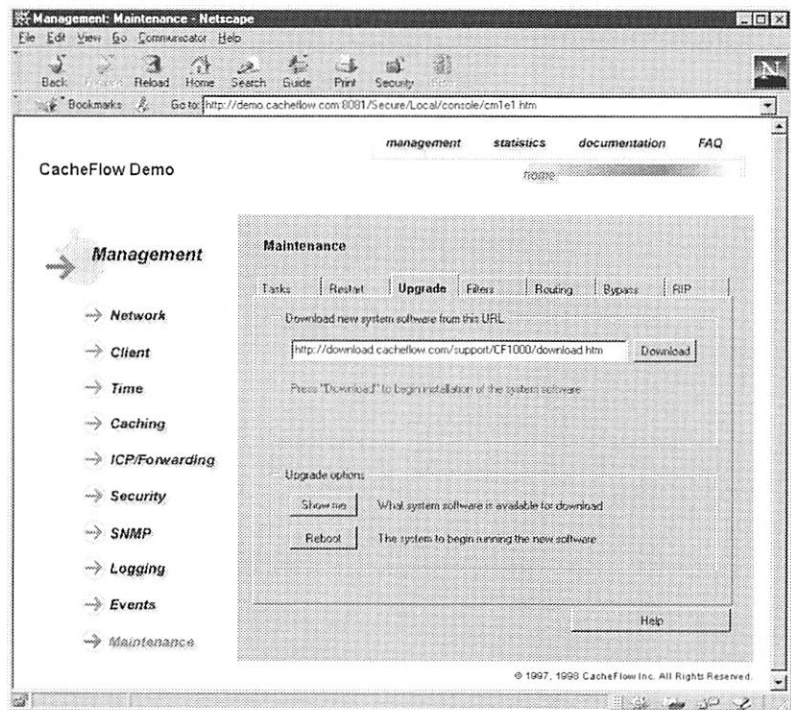


Figure 6-18

Upgrading the System

To upgrade the CacheOS you must specify the URL where the new system files are located.

Displaying the Event Log

The event log contains a record of system events that occur on the CacheOS. To display the event log, follow the steps outlined below:

1. Select Statistics from the CacheOS home page.
2. Select the Event Viewer applet.
3. Click the forward arrow and back arrow buttons to move through the event list.

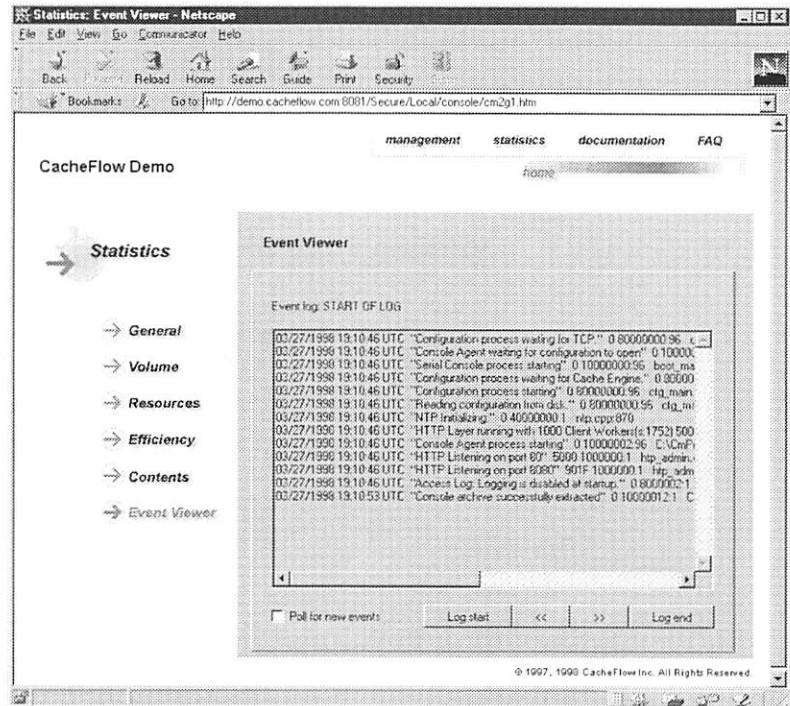


Figure 6-20
Displaying the Event Log
The Event Viewer applet displays the event log.

Using Syslog Monitoring

Syslog is an event monitoring scheme that is especially popular in UNIX environments. In sites where Syslog is used, there is typically a “log host” node, which acts as a “sink” for several devices on the network. You must have a Syslog daemon operating in your network.

Syslog format: “Date Time Hostname Event”

To configure syslog, follow the steps outlined below:

1. Select Management from the CacheOS home page.
2. Select the Events applet.
3. Click the Syslog tab.
4. Configure Syslog with the address “log host.”
5. Check the Enable Syslog checkbox.
6. To save your changes to the device, click the Apply button.

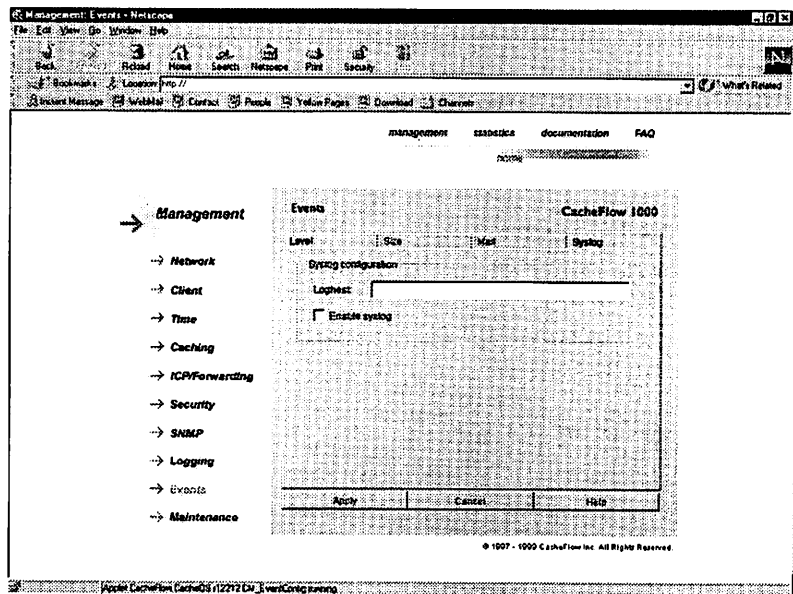


Figure 6-21
Setting up Syslog
monitoring

*The Events applet displays
the Syslog tab.*

Using Name Imputing

Name imputing allows the CacheOS to resolve host names based on a partial name specification. When the CacheOS submits a host name to the DNS server, the DNS server resolves the name to an IP address. If the host name cannot be resolved, the CacheOS adds the first entry in the name imputing list to the end of the host name and resubmits it to the DNS server. The CacheOS tries each entry in the name imputing list until the name is resolved, or the end of the list is reached. If the end of the list is reached and the name is not resolved, the CacheOS returns a failure.

For example, if the name imputing list contains the following entries

berkeley.edu

edu

and a user submits a host name of "eedept" the CacheOS will try to resolve the following host names in the order shown:

eedept

eedept.berkeley.edu

eedept.edu

To add names to the imputing list, follow the steps outlined below:

1. Select Management from the CacheOS home page.
2. Select the Imputing tab in the Network applet.
3. Click the New button to add a new name to the imputing list.
4. Enter the name and click the Ok button.
5. Click the Apply button to save your changes.

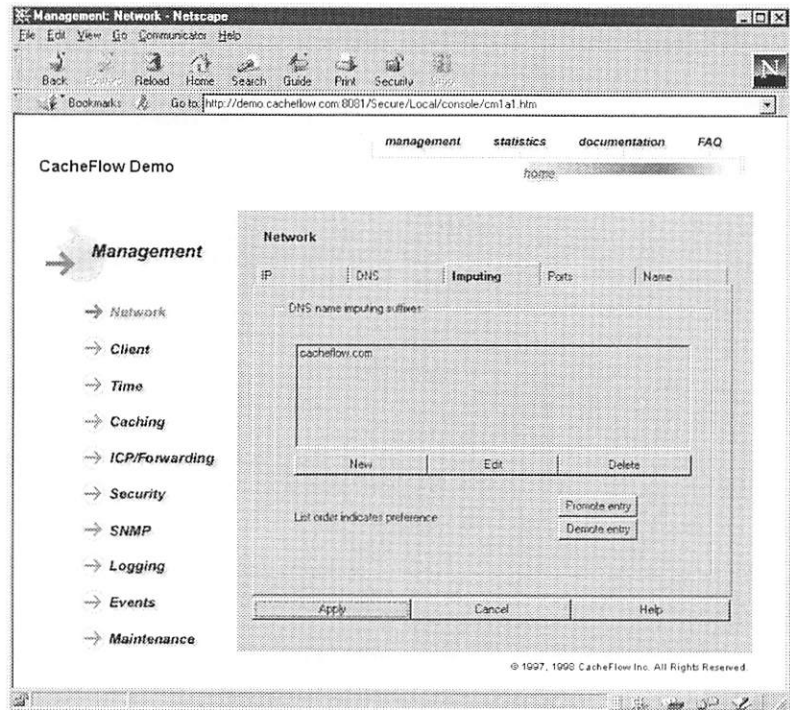


Figure 6-22
Using Name Imputing

The CacheOS can use name imputing to resolve partial host names.

Configuring Cache Options

When the CacheFlow device retrieves an object from the Web and returns it to the client, the object is considered fresh--the Web cache knows it is fresh because it just retrieved the object from the source. The goal of the Web cache is to keep fresh as many of the objects in the cache as possible, so when the objects are requested the CacheOS can deliver them to the client without having to retrieve them from the source.

To keep objects fresh, the Web cache uses a variety of techniques to learn the update patterns of the objects. It then refreshes the objects in the background, before they are requested by a client. You can define how hard the Web cache works to ensure freshness. If you set the desired freshness to 99%, the Web cache will work harder verifying objects are fresh than if you set the desired freshness to 95%. The higher you set the desired freshness, the higher the percentage of objects will be fresh in the cache when they are requested.

To set the desired freshness, follow the steps outlined below:

1. Select Management from the CacheOS home page.
2. Select the Caching applet.
3. Enter the desired freshness in the Desired freshness field, up to 100%.
4. If you want to limit the network bandwidth the Web cache can use to maintain freshness, select the radio button next to the Limit refresh bandwidth field and enter the bandwidth limit. You should let the CacheOS manage refresh bandwidth unless you are experiencing problems on your network.
5. To save your changes to the device, click the Apply button.

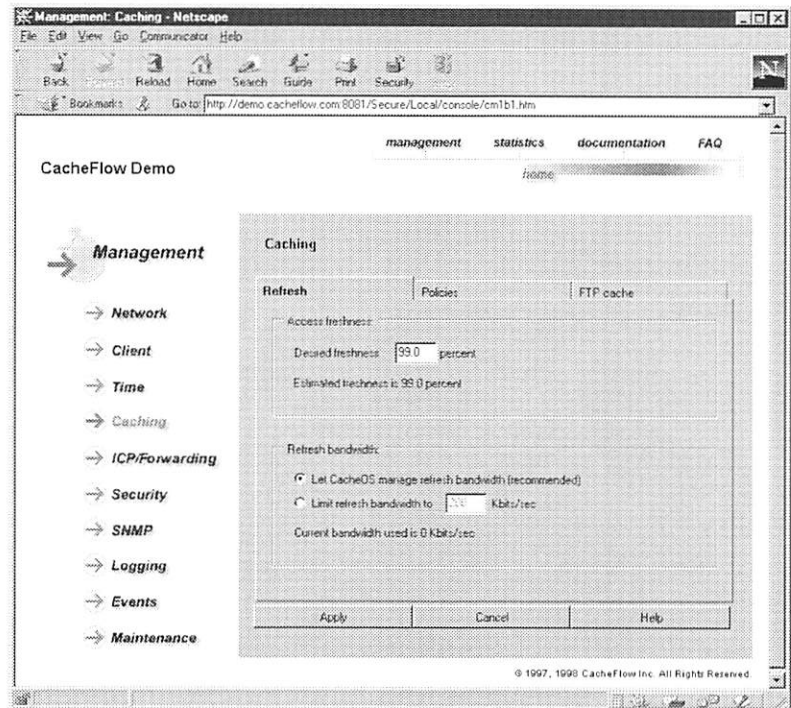


Figure 6-23
Setting Caching Options
The caching options are set
on the Refresh tab in the
Caching applet.

Setting the Network Bandwidth Utilization

You can configure the amount of bandwidth the CacheOS uses for refresh operations. The CacheOS tracks the amount of time it takes to retrieve an object, and calculates the bandwidth utilization based on past performance. You set the bandwidth utilization in Kbits per second. For example, if your network bandwidth to the outside world is 256 Kbits per second, setting the bandwidth utilization to 25 Kbits per second will restrict the CacheOS to approximately 10% of the network bandwidth for background refresh operations. If you have multiple Web caches, you must take into account each cache when setting the bandwidth utilization. In the example above, if you had two CacheFlow Web caches you would set the bandwidth utilization to 12 Kbits for each cache to maintain 10% utilization across both caches.

Setting Policies

When an object in the cache expires, it is placed in a refresh list. The CacheOS processes the refresh list in the background, when it is not serving requests. The refresh policies define how the CacheOS handles the refresh process.

To set the refresh policies, follow the steps outlined below:

1. Select Management from the CacheOS home page.
2. Select the Caching applet.
3. Select the Policies tab.
4. Select the refresh options you want to use.
5. Click the Apply button to save your changes.

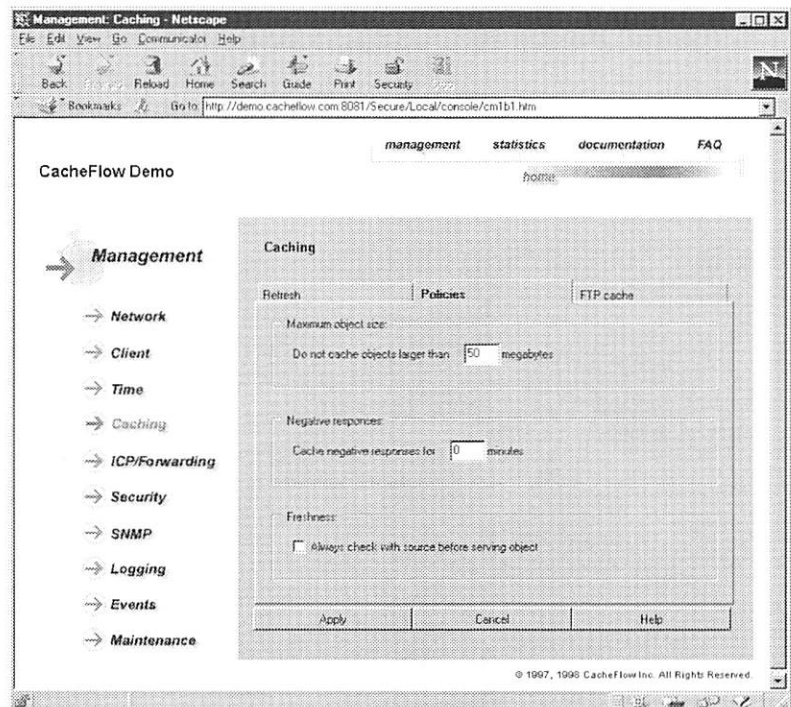


Figure 6-24
Setting Policies

The refresh policies control how the CacheOS manages refresh operations.

Setting the Maximum Object Size

You can set the maximum object size to store in the cache. All objects retrieved that are greater than the maximum object size will be delivered to the client, but they will not be stored in the Cache.

Caching Negative Responses

If the CacheOS receives a failed response when attempting to retrieve an object, it can cache the negative response. If the CacheOS caches the response, it will return a failure for additional requests for the specified number of minutes. If the CacheOS does not cache the response, it will continue to attempt each failed request.

Guaranteed Freshness

The Freshness option allows you to guarantee that all objects served from the cache are current. When this option is checked, the Web cache will check the object on the source before serving it to the client. This option helps increase the accuracy of objects served, but it also slows down every request because the CacheOS must verify each object before it sends it to the client.

Setting FTP Caching Options

In addition to HTTP objects, the CacheOS can cache FTP requests. When the CacheOS retrieves an FTP object and places it in the cache, it must have a way to determine how long the object should stay in the cache. There are two options:

- If the object has a last modified date, the CacheOS will assign a refresh date to the object that is some percentage of the last modified date.
- If the object does not have a last modified date, the CacheOS can assign a refresh date to the object based on a fixed period of time.

The FTP caching options allow you to specify the maximum size of FTP objects to cache, and how long objects should stay in the cache. To configure the FTP cache options, follow the steps outlined below:

1. Select Management from the CacheOS home page.
2. Select the Caching applet.
3. Select the FTP caching tab.

4. Update the FTP caching options.
5. Click the Apply button to save your changes.

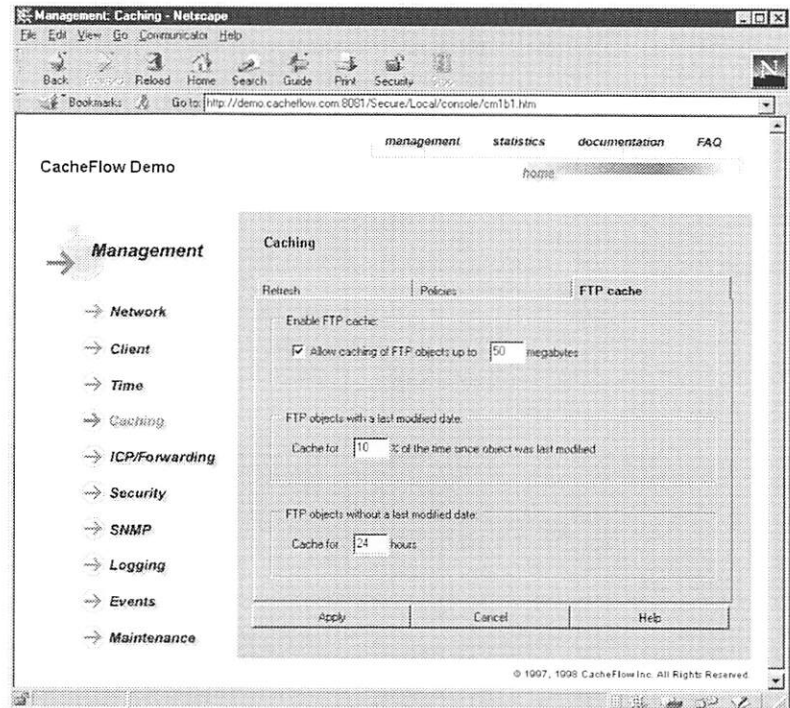


Figure 6-25
FTP Caching

The FTP caching options allow you to configure how long FTP objects are stored in the cache.

Using ICP and Advanced Forwarding

See "Configuring Hierarchical Caches" on page 17 for detailed information on using ICP and advanced forwarding options.

ICP (Internet Caching Protocol) and advanced forwarding allow you to forward requests not found in the cache to a cache hierarchy. ICP is a communication protocol that allows the cache to locate objects from neighboring cache hosts. Advanced forwarding allows you to create complex forwarding schemes. With advanced forwarding you can forward requests based on the URL or IP address of the requested object, and you can define multiple forwarding hosts and groups of forwarding hosts. For detailed information on creating ICP and advanced forwarding configurations, see "Configuring Hierarchical Caches" on page 17.

Loading the Advanced Forwarding and ICP Configuration

For detailed information on creating ICP and advanced forwarding configurations, see "Configuring Hierarchical Caches" on page 17. Once you have created the ICP or advanced forwarding configuration file, place the file on an HTTP server so it can be downloaded to the CacheFlow Web cache. To download the list to the CacheFlow Web cache follow the steps outlined below:

1. Select Management from the CacheOS home page.
2. Select the ICP/Forwarding applet.
3. Enter the URL where the configuration file is located.
4. Click the Install button to download the configuration file.

You can click the View button to display the configuration file before installing it.

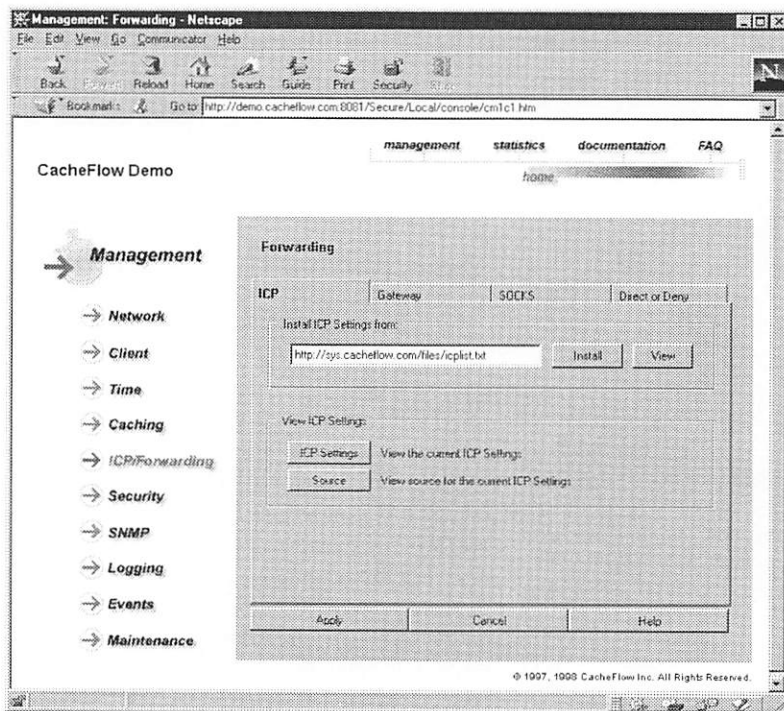


Figure 6-26
Configuring ICP

The ICP configuration file defines the ICP hosts in your cache hierarchy.

Configuring WCCP Support

The CacheFlow device can be configured to participate in a WCCP (Web Cache Control Protocol) scheme, where a WCCP-capable router collaborates with a set of WCCP-configured CacheFlow devices to service requests. WCCP is a Cisco-developed protocol. For more information about WCCP, refer to the *Appendix B: WCCP (Web Cache Control Protocol)*.

To configure the CacheFlow device for WCCP, follow the steps outlined below:

1. Select Management from the CacheOS home page.
2. Select the Forwarding applet.
3. Select the WCCP tab.
4. Enter the URL where the configuration file is located.
5. Click the Install button to download the configuration file.

You can click the View button to display the configuration file before installing it.

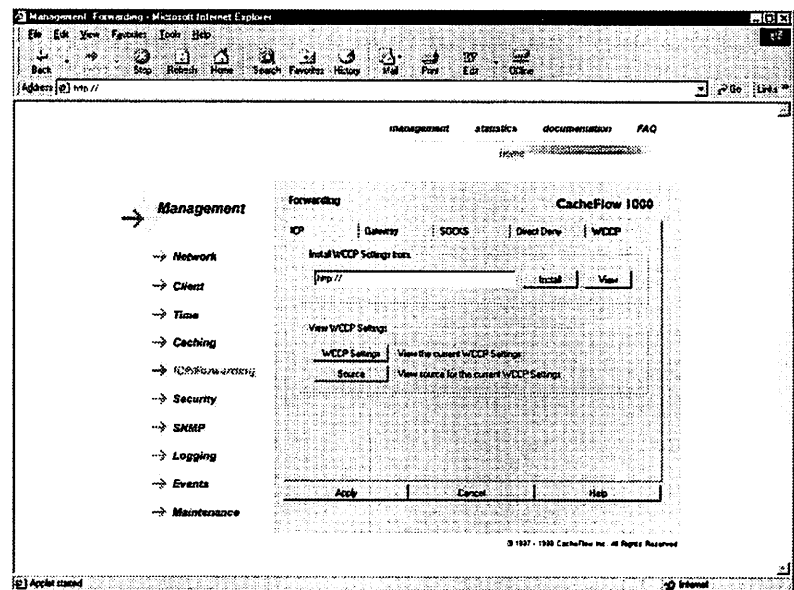


Figure 6-27
Configuring WCCP

Using Simple Forwarding

See "Configuring Hierarchical Caches" on page 17 for detailed information on using forwarding options.

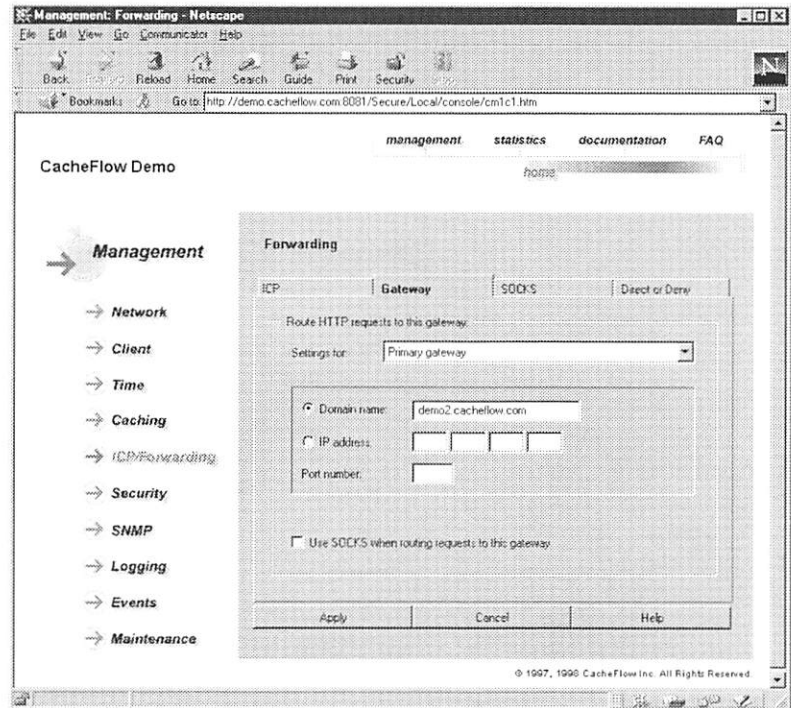
Simple forwarding allows you to define a forwarding gateway the CacheOS will use to forward all requests not found in the cache. If a gateway is specified, when an object is requested that is not in the cache the Web cache will forward the request to the gateway rather than retrieve the object from the network. A primary and alternate gateway can be specified. For information on using simple forwarding, see "Configuring Hierarchical Caches" on page 17.

To configure a forwarding gateway, follow the steps outlined below:

1. Select Management from the Web cache home page.
2. Select the Forwarding applet.
3. Select the Gateway tab.
4. Select the gateway to configure (primary or alternate) in the Settings for drop-down list.
5. Enter the domain name or IP address and port.
6. If the gateway is a SOCKS server, enable the Use SOCKS when forwarding requests to this gateway checkbox.
7. Click the Apply button to save your changes.

Figure 6-28
Configuring Forwarding Options

The Forwarding applet allows you to specify a primary and alternate gateway to use for objects not stored in the cache.



Using a SOCKS Server

If you are using a SOCKS server for the primary or alternate forwarding gateway, you must specify the CacheFlow Web cache's machine ID for the Identification (Ident) protocol used by the SOCKS gateway.

To specify the machine ID, follow the steps outlined below:

1. Select Management from the CacheOS home page.
2. Select the Forwarding applet.
3. Select the SOCKS tab.
4. Enter the machine ID in the Machine ID field.
5. Click the Apply button to save your changes.

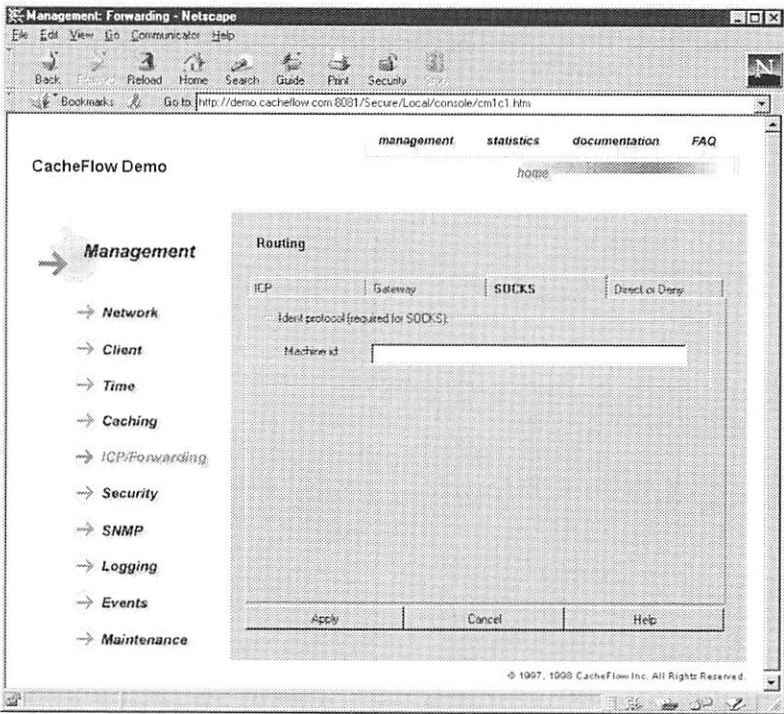


Figure 6-29
Using a SOCKS Server

To use a SOCKS server as the primary or alternate gateway you must specify the machine ID for the CacheFlow Web cache.

Creating a Direct or Deny List

When using a proxy gateway, the CacheOS forwards requests to the gateway server. The gateway server then determines what to do with the request. The gateway can be used to forward requests to an external network.

Direct addresses are addresses the CacheOS should send out on the network rather than forward to the gateway. Deny addresses are addresses to which the CacheOS should deny access. The direct and deny address specifications are made up of a subnet and mask. Requested addresses are compared to the subnet and mask to determine a match. If the request does not match an address in the direct or deny list, the CacheOS sends the request to the gateway.

The direct or deny list is a simple text file containing a list of IP addresses, subnet masks, and commands. A sample direct or deny list is illustrated below:

206.94.0.0	255.255.0.0	DENY
206.96.0.0	255.255.0.0	DENY
206.65.0.0	255.255.0.0	DIRECT

To enter a direct or deny list, create a text file with the direct or deny commands then place the file on an HTTP server. To download the list to the CacheOS follow the steps outlined below:

1. Select Management from the CacheOS home page.
2. Select the Forwarding applet.
3. Select the Direct or Deny tab.
4. Enter the URL where the Direct or Deny list is located.
5. Click the Install button to download the list.

You can click the View button to display the list before installing it.

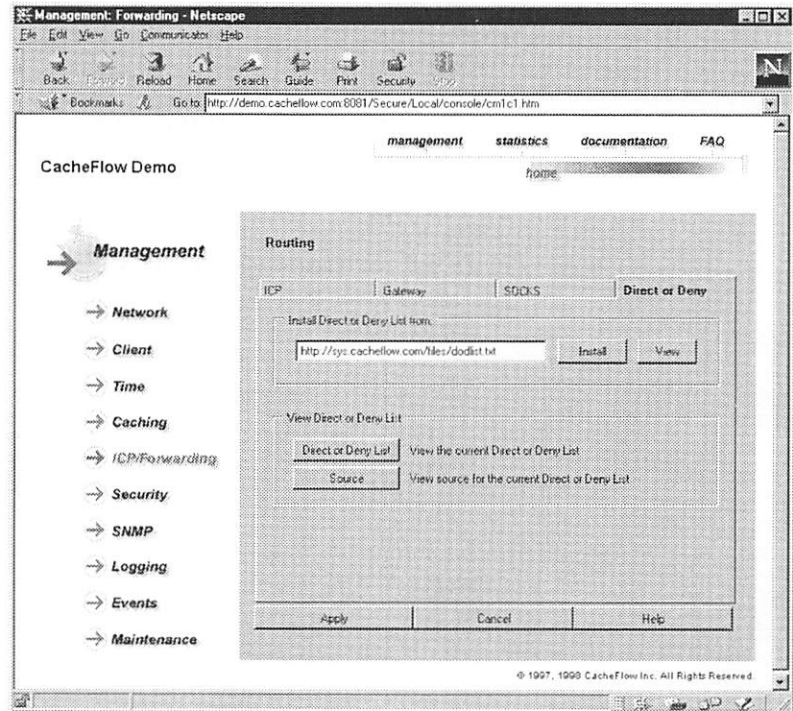


Figure 6-30
Creating a Direct or Deny List

The direct or deny list controls how requests are routed to the gateway.

Using a Filter List

The CacheOS can filter requests made by clients using a filter list. When a filter list is loaded, all requested URLs are compared to the list, and processed based on the results.

The filter list can be used to assign the following actions for a URL:

- deny service
- access direct
- do not refresh
- do not cache
- cache advertising objects
- case-insensitive matching

To use the filter list, create a text file that contains a filter statement for each URL you want to filter. The syntax of the filter statement is shown below:

url filter = value

The URL can be specified as a full URL, or a regular expression. For information on using regular expressions, see "Regular Expressions" on page 173. The URL can be a host name or directory path. Some sample URLs are shown below:

www.yahoo.com

www.yahoo.com/quotes/

www.yahoo.com/news

If you specify a directory path, the CacheOS will treat the specification as a full directory name. For example, the URL www.yahoo.com/news will not match the URL www.yahoo.com/newspaper.

The filters are described below:

Parameters:	Value	Description
service	no	Deny service to the URL.
direct	yes	Do not forward requests to a parent proxy or SOCKS server. This filter only applies when the device is configured to forward requests.
refresh	no	Do not refresh the object if it is cached.
cache	no	Do not cache the object.
advertisement	yes	Cache objects at this URL, and request the objects in the background to maintain the hit count.
case_insensitive	yes	Match URLs using case-insensitivity. By default all URLs are matched in a case-sensitive manner. This filter should be set to match URLs served by Operating Systems such as Windows NT, which is case insensitive.

You can specify multiple filters in a single filter statement. Sample filter statements are shown below:

```
ftp://.* direct=yes
http://www.dilbert.com service=no
http://www.nytimes.com cache=no refresh=no
http://www.engservices.com direct=yes
http://.*\cache\flow.* direct=yes
http://www.msnbc.com case_insensitive=yes cache=no
```

Type each statement on a separate line. You can use the ";" character at the beginning of a line to enter comments. The maximum length of a line is 4096 bytes.

Using the Filter List to Restrict Access

You can also use the filter list to restrict user access to the CacheOS. You can create an ACL (access control list) that defines the IP addresses and subnets that have access to the CacheOS, then eliminate service to all addresses not in the list. The commands required to create an ACL are shown below:

```
# create an ACL named myusers
define acl myusers
    192.168.24.0/24
end acl myusers

# eliminate service to all users not in the myusers ACL
ALL acl=myusers service=no cache=no
```

This example creates an ACL named "myusers" that contains all addresses on the subnet of 192.168.24.0. It then sets the service and cache options to "no" for anyone not in the list (the exclamation point "!" negates the list). Note that the syntax is case sensitive, and the ALL rule must be in uppercase.

Creating Filter Lists

There are two types of filter lists: the local filter list, and the central filter list.

Local Filter List

The local filter list is a list you create and maintain on your network. You can use a local filter list alone, or in conjunction with a central list.

Once the local filter list is created, place it on an HTTP server so it can be downloaded. To download the filter list to the CacheFlow Web cache, follow the steps outlined below:

1. Select Management from the CacheOS home page.
2. Select the Maintenance applet.
3. Select the Filters tab.
4. Enter the URL where the filter list is located in the Local file field.
5. Click the Install button to download the list.

You can click the View button to display the list before installing it.

Central Filter List

The central filter list is a shared list of addresses that is used by multiple CacheFlow devices. You can create your own central filter list to manage multiple CacheFlow devices, or you can use the central filter list maintained by CacheFlow Technical Support at the following URL:

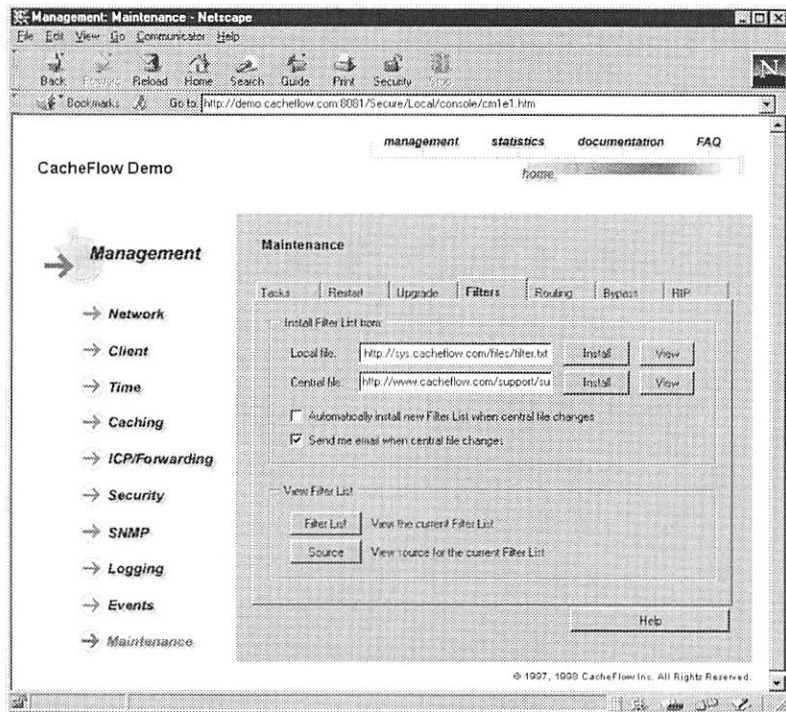
<http://www.cacheflow.com/support/subscriptions/CentralFilterList.txt>

Once the central filter list is created, place it on an HTTP server so it can be downloaded. To download the filter list to the CacheFlow Web cache, follow the steps outlined below:

1. Select Management from the CacheOS home page.
2. Select the Maintenance applet.
3. Select the Filters tab.
4. Enter the URL where the filter list is located in the Central file field.
5. Click the Install button to download the list.

You can click the View button to display the list before installing it.

Figure 6-31
Using Filter Lists
The filter list is used to filter
HTTP requests.



Defining Static Routes

The CacheFlow device can be configured to use static routes. To use static routes you must create a routing table and place it on an HTTP server accessible to the CacheFlow device. The routing table is a text file that contains a list of IP addresses, subnet masks, and gateways. When you download a routing table, the table is stored in the device until it is replaced by downloading a new table.

The routing table is a simple text file containing a list of IP addresses, subnet masks, and gateways. A sample routing table is illustrated below:

206.63.0.0	255.255.0.0	206.63.158.213
206.64.0.0	255.255.0.0	206.63.158.213
206.65.0.0	255.255.0.0	206.63.158.226

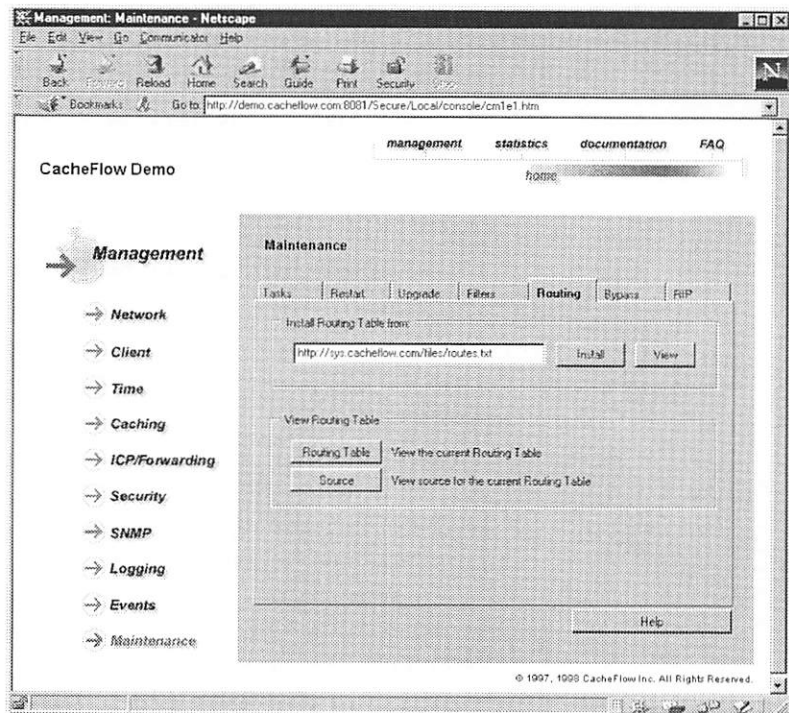
When a routing table is loaded, all requested addresses are compared to the list, and routed based on the best match.

Once the routing table is created, place it on an HTTP server so it can be downloaded to the device. To download the routing table to the CacheOS, follow the steps outlined below:

1. Select Management from the CacheOS home page.
2. Select the Maintenance applet.
3. Select the Routing table tab.
4. Enter the URL where the routing table is located.
5. Click the Install button to download the table.

You can click the View button to display the routing table before installing it.

Figure 6-32
Defining Static Routes
The routing table is used to
route requests.



Using a Bypass List

The bypass list is only used for transparent caching.

Bypass routes are used to prevent the CacheFlow device from transparently proxying requests to servers that perform IP authentication with clients. The bypass list contains a list of IP addresses, subnet masks, and gateways. When a request matches an IP address and subnet mask specification in the bypass list, the request is sent to the designated gateway.

To use bypass routes, you must create a text file that contains a list of address specifications. The file should be named with a .txt extension, and it can contain no HTML. Once you have created the bypass list, place it on an HTTP server so it can be downloaded to the CacheFlow device.

There are two types of bypass lists: the local bypass list, and the central bypass list.

Local Bypass List

The local bypass list is a list you create and maintain on your network. You can use a local bypass list alone, or in conjunction with a central list.

The gateways specified in the bypass list must be on the same subnet as the CacheOS device.

The local bypass list contains a list of IP addresses, subnet masks, and gateways. It can also define the default bypass gateway to be used by both the local bypass list and central bypass list. The gateways specified in the bypass list must be on the same subnet as the CacheFlow device. When you download a bypass list, the list is stored in the device until it is replaced by downloading a new list. A sample local bypass list is shown below:

```

;define the default gateway for the local and central bypass list
BYPASS_GATEWAY 192.168.10.20

;define addresses to bypass
;IP address      subnet      gateway (or use default gateway)
192.168.10.218   255.255.255.255
192.168.10.120   255.255.255.255
172.16.0.0       255.255.255.0    192.168.10.22

```

If you do not specify the BYPASS_GATEWAY, and you do not designate the gateway in the address specification, the CacheFlow device will forward the request to the default gateway defined by the network configuration.

Once the bypass list is created, place it on an HTTP server so it can be downloaded to the device. To download the bypass list, follow the steps outlined below:

1. Select Management from the CacheOS home page.
2. Select the Maintenance applet.
3. Select the Bypass tab.
4. Enter the URL where the bypass list is located in the Local file field.
5. Click the Install button to download the list.

You can click the View button to display the list before installing it.

Central Bypass List

The central bypass list is a shared list of addresses that is used by multiple CacheFlow devices. The central list contains addresses to bypass, but does not specify gateways (because the CacheFlow devices will be located on different subnets, using different gateways). The gateway used for matches in the central bypass list is defined using the `BYPASS_GATEWAY` command in the local bypass list. If there is no `BYPASS_GATEWAY` command, the CacheFlow device will use the default gateway defined by the network configuration.

You can create your own central bypass list to manage multiple CacheFlow devices, or you can use the central bypass list maintained by CacheFlow Technical Support at the following URL:

<http://www.cacheflow.com/support/subscriptions/CentralBypassList.txt>

The central bypass list maintained by CacheFlow contains addresses CacheFlow has identified as using client authentication.

Once the bypass list is created, place it on an HTTP server so it can be downloaded to the device. To download the bypass list, follow the steps outlined below:

1. Select Management from the CacheOS home page.
2. Select the Maintenance applet.
3. Select the Bypass tab.
4. Enter the URL where the bypass list is located in the Central file field.
5. Click the Install button to download the list.

You can click the View button to display the list before installing it.

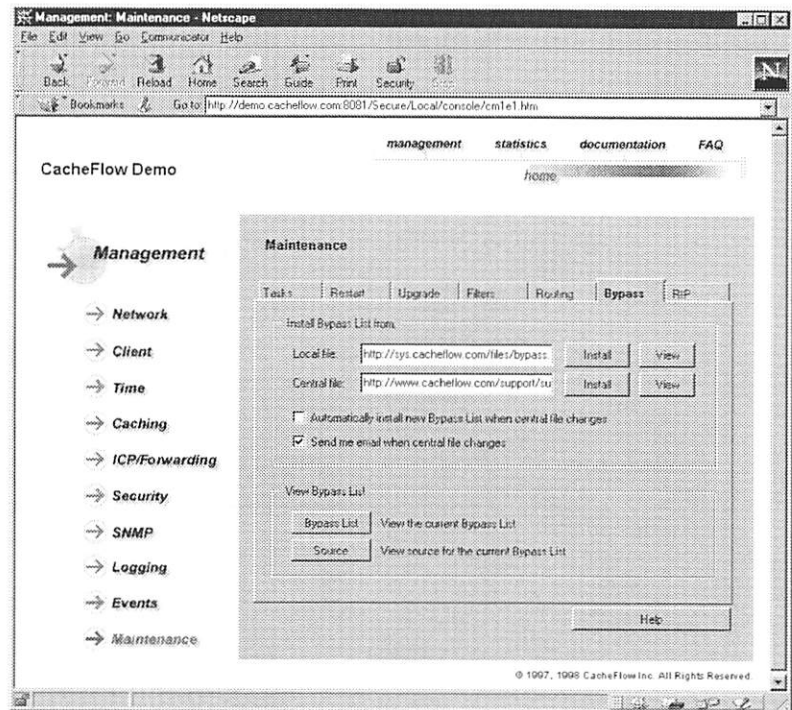


Figure 6-33

Using a Bypass List

The bypass list is used with transparent caching to support servers that perform IP authentication.

Using RIP

The RIP configuration is defined in a configuration file. To configure RIP, first create a text file of RIP commands, then load the file on the CacheFlow web cache. The RIP commands are described below:

net

net Nname[/mask] gateway Gname metric Value <passivelactiveexternal>

Syntax

Parameters:	Description
Nname	Name of the destination network. It can be a symbolic network name, or an Internet address specified in dot notation.
/mask	Optional number between 1 and 32 indicating the netmask associated with Nname.
Gname	Name or address of the gateway to which RIP responses should be forwarded.
Value	The hop count to the destination host or network. A net Nname/32 specification is equivalent to the host Hname command.
passivelactiveexternal	Indicates whether the gateway should be treated as passive or active, or whether the gateway is external to the scope of the RIP protocol.

host

host Hname gateway Gname metric Value <passive|active|external>

Syntax

Parameters:	Description
Hname	Name of the destination network. It can be a symbolic network name, or an Internet address specified in dot notation.
Gname	Name or address of the gateway to which RIP responses should be forwarded. It can be a symbolic network name, or an Internet address specified in dot notation.
Value	The hop count to the destination host or network. A net Nname/32 specification is equivalent to the host Hname command.
passive active external	Indicates whether the gateway should be treated as passive or active, or whether the gateway is external to the scope of the RIP protocol.

Parameters

Lines that do not start with net or host commands, must consist of one or more of the following parameter settings, separated by commas or blanks:

Parameters:	Description
if=[0 1 2 3]	Indicates that the other parameters on the line apply to the interface numbered 0,1,2, or 3, in CacheOS terms.
passwd=XXX	Specifies a RIPv2 password that will be included on all RIPv2 responses sent and checked on all RIPv2 responses received. The password must not contain any blanks, tab characters, commas or '#' characters.
no_ag	Turns off aggregation of subnets in RIPv1 and RIPv2 responses.
no_super_ag	Turns off aggregation of networks into supernets in RIPv2 responses.

Parameters:	Description
passive	Marks the interface to not be advertised in updates sent via other interfaces, and turns off all RIP and router discovery through the interface.
no_rip	Disables all RIP processing on the specified interface.
no_ripv1_in	Causes RIPv1 received responses to be ignored.
no_ripv2_in	Causes RIPv2 received responses to be ignored.
ripv2_out	Turns off RIPv1 output and causes RIPv2 advertisements to be multicast when possible.
ripv2	Is equivalent to no_ripv1_in and no_ripv1_out. This parameter is set by default.
no_rdisc	Disables the Internet Router Discovery Protocol. This parameter is set by default.
no_solicit	Disables the transmission of Router Discovery Solicitations.
send_solicit	Specifies that Router Discovery solicitations should be sent, even on point-to-point links, which by default only listen to Router Discovery messages.
no_rdisc_adv	Disables the transmission of Router Discovery Advertisements.
rdisc_adv	Specifies that Router Discovery Advertisements should be sent, even on point-to-point links, which by default only listen to Router Discovery messages.
bcast_rdisc	Specifies that Router Discovery packets should be broadcast instead of multicast.
rdisc_pref=N	Sets the preference in Router Discovery Advertisements to the integer N.
rdisc_interval=N	Sets the nominal interval with which Router Discovery Advertisements are transmitted to N seconds and their lifetime to 3*N.

Parameters:	Description
<code>trust_gateway=rname</code>	Causes RIP packets from that router and other routers named in other <code>trust_gateway</code> keywords to be accept, and packets from other routers to be ignored.
<code>redirect_ok</code>	Causes RIP to allow ICMP Redirect messages when the system is acting as a router and forwarding packets. Otherwise, ICMP Redirect messages are overridden.

CacheOS Specific Parameters

The following RIP parameters are unique to the CacheOS configuration:

Parameters:	Description
<code>no_rip_out</code>	Disables the transmission of all RIP packets. This parameter is set by default.
<code>no_ripv1_out</code>	Disables the transmission of RIPv1 packets.
<code>no_ripv2_out</code>	Disables the transmission of RIPv2 packets.
<code>rip_out</code>	Enables the transmission of RIPv1 packets.
<code>ripv1_out</code>	Enables the transmission of RIPv1 packets.
<code>rdisc</code>	Enables the transmission of Router Discovery Advertisements.
<code>ripv1</code>	Causes RIPv1 packets to be sent.
<code>ripv1_in</code>	Causes RIPv1 received responses to be handled.

Using Passwords

The first password specified for an interface is used for output. All passwords pertaining to an interface are accepted on input. For example,

```
if=0 passwd=aaa  
if=1 passwd=bbb  
passwd=ccc
```

Interface 0 would accept passwords aaa and ccc, and would transmit using password aaa. Interface 1 would accept passwords bbb and ccc, and would transmit using password bbb. The other interfaces would accept and transmit the password ccc.

Configuring RIP

Once the RIP configuration is created, place it on an HTTP server so it can be downloaded to the device. To download the RIP configuration, follow the steps outlined below:

1. Select Management from the CacheOS home page.
2. Select the Maintenance applet.
3. Select the Rip tab.
4. Enter the URL where the RIP configuration file is located.
5. Click the Install button to download the configuration file.

You can click the View button to display the configuration file before installing it.

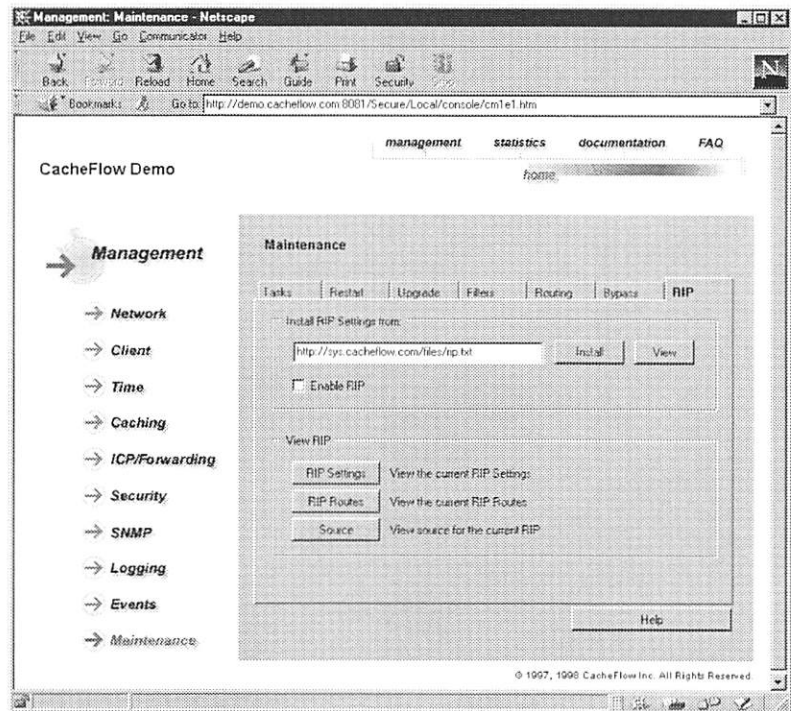


Figure 6-34
Using RIP

The RIP configuration can be loaded on the CacheOS device using the Web interface or the command line interface.

Displaying the System Status

To view the system status, follow the steps outlined below:

1. Select Statistics from the CacheOS home page.
2. The system status is displayed in the General applet.
3. Select the Environment tab to display the current hardware status.
4. Select the Disks tabs to display information on the installed disks.

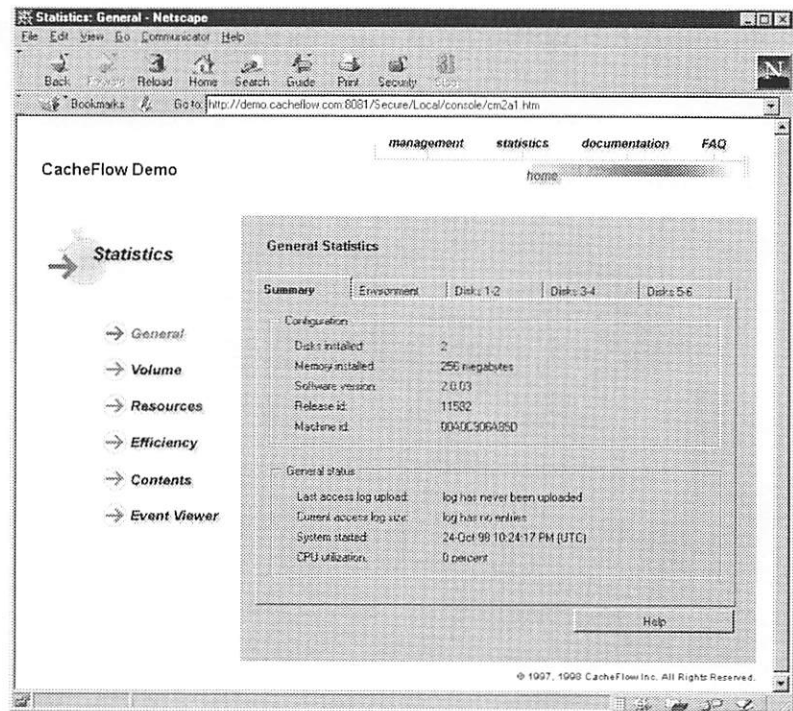


Figure 6-35
Displaying the
System Status

The system status is displayed in the General applet.

Displaying the Number of Objects and Bytes Served

To view the number of objects and bytes served, follow the steps outlined below:

1. Select Statistics from the CacheOS home page.
2. Select the Volume applet.
3. The Objects tab shows the number of objects served over the last 60 minutes, 24 hours, and 30 days.
4. The Bytes tab shows the number of bytes served over the last 60 minutes, 24 hours, and 30 days.

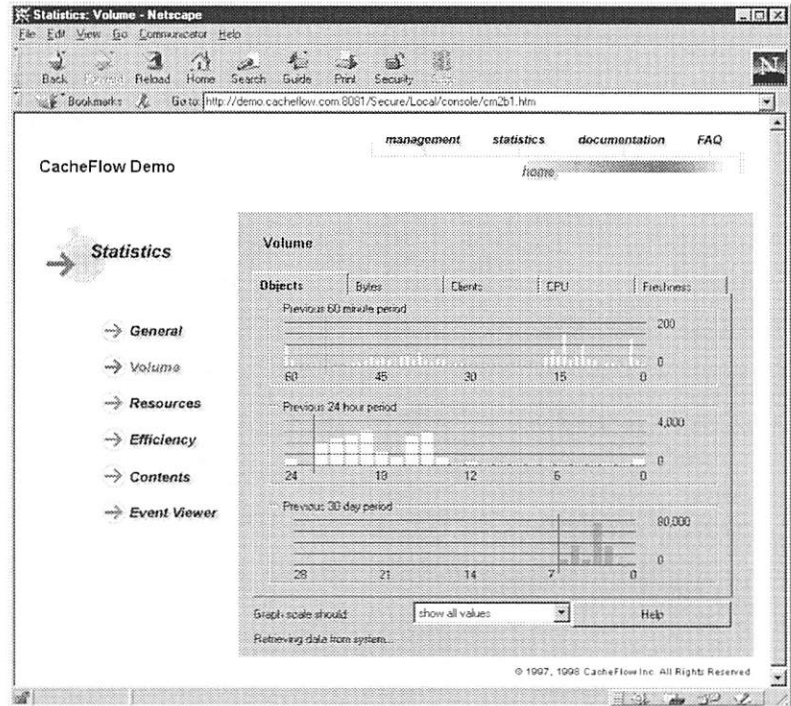


Figure 6-36
**Displaying the Number of
Objects and Bytes Served**

*The Volume applet displays
the volume over the last 60
minutes, 24 hours,
and 30 days.*

Displaying Active Client Connections

To view the number of active clients, follow the steps outlined below:

1. Select Statistics from the CacheOS home page.
2. Select the Volume applet.
3. Select the Clients tab.

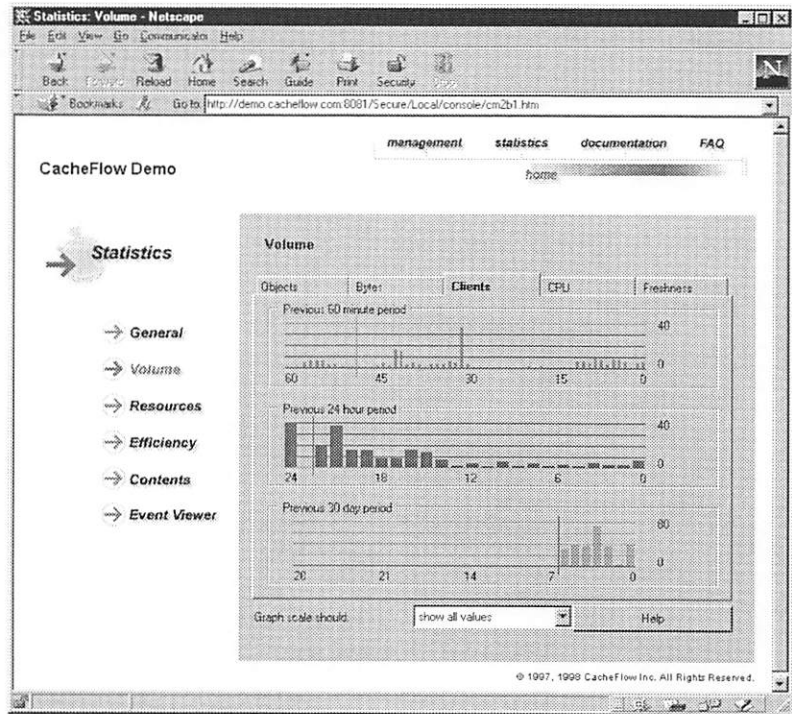


Figure 6-37

Displaying Active Clients

The Volume applet displays the active clients over the last 60 minutes, 24 hours, and 30 days.

Displaying How Resources are Currently Being Used

To view how the CacheOS disk and memory resources are being used, follow the steps outlined below:

1. Select Statistics from the CacheOS home page.
2. Select the Resources applet.
3. The Disk use tab shows how disk resources are shared by the cache, system objects, and access log.
4. The Memory use tab shows memory resources are shared by the cache, the system, and network buffers.
5. The Data tab lists totals for the disk and memory resources.

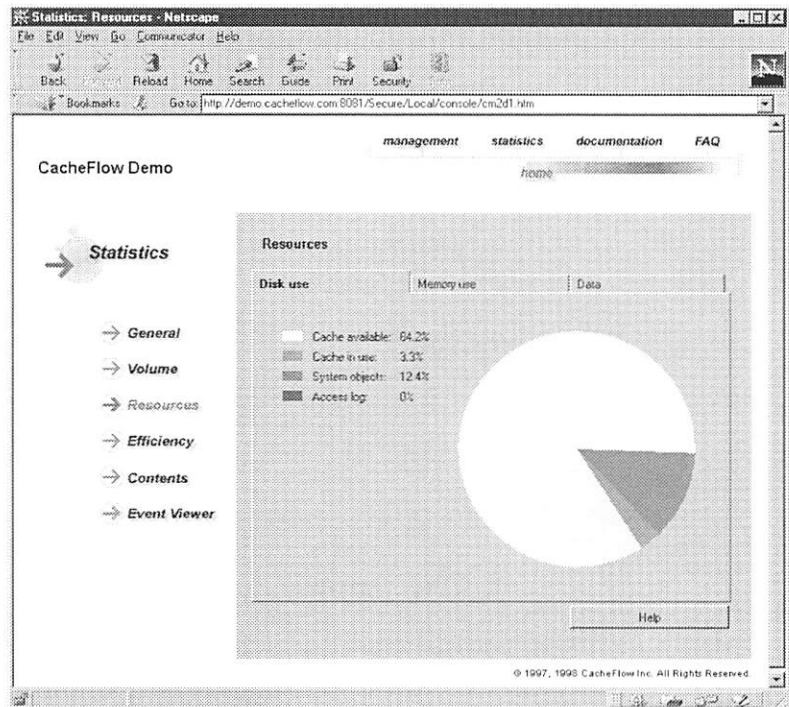


Figure 6-38
Displaying the Disk and Memory Resources

The Resources applet shows how the CacheFlow Web cache's disks and memory are used.

Displaying Cache Efficiency

To view the cache efficiency, follow the steps outlined below:

1. Select Statistics from the CacheOS home page.
2. Select the Efficiency applet.
3. The Summary tab shows the percent of objects served from cache, the percent loaded from the network, and the percent that were non-cacheable.
4. The Non-cacheable tab shows the breakdown of non-cacheable objects.
5. The Access pattern tab shows the percent of objects accessed from RAM and the percent of objects access from disk.
6. The Data tab lists totals for objects served.

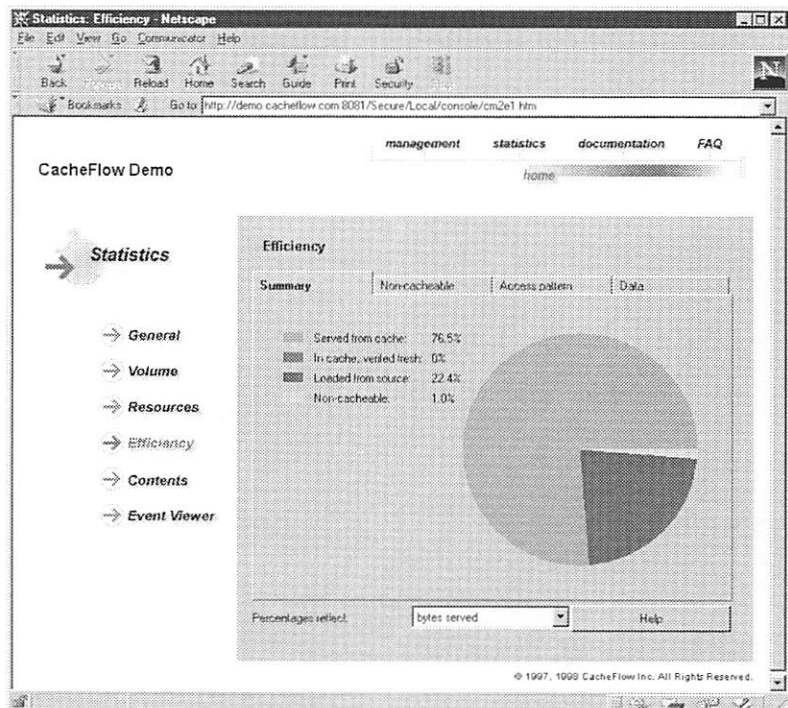


Figure 6-39
Displaying Cache Efficiency

The Efficiency applet shows how objects were served from the cache.

Displaying the Cache Contents Distribution

The cache contents include all objects currently stored on the CacheOS. The cache contents are not cleared when the CacheFlow Web cache is powered off. To view the distribution of cache contents, follow the steps outlined below:

1. Select Statistics from the CacheOS home page.
2. Select the Contents applet.
3. The Distribution tab shows the number of objects stored in the cache by size.
4. The Data tab lists totals for the number of objects served by size.

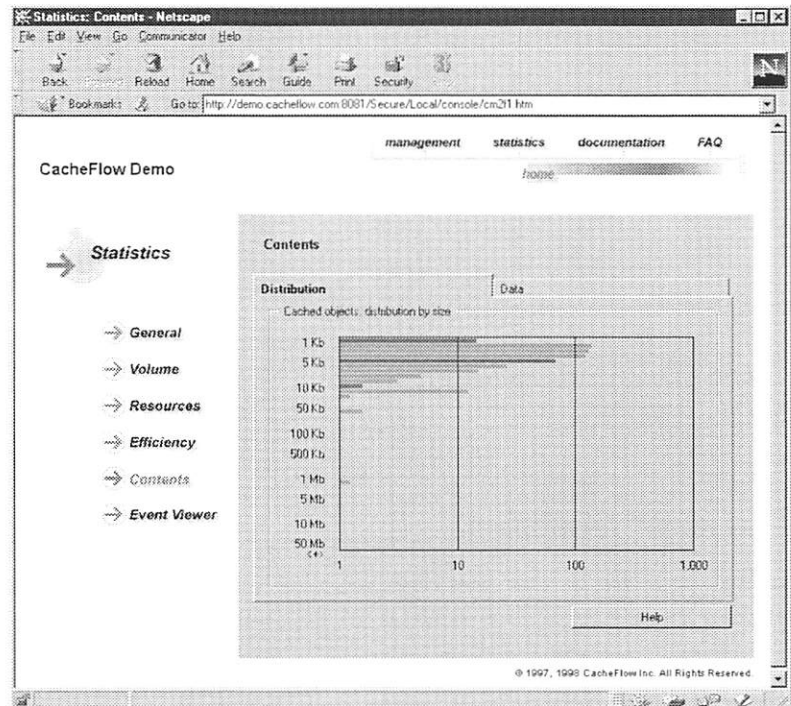


Figure 6-40
Displaying the Cache Contents

The Contents applet shows all objects currently stored in the cache.

Appendix A

Access Log Formats

Overview

The CacheFlow device can create access logs in one of three formats: the common log format, Squid-compatible format, or custom format. Each format is described below.

Common Access Log Format

The common log format contains one line for each request. The format of each log entry is shown below:

```
remotehost rfc931 authuser [date] "request" status bytes
```

Each field is described below:

Field Name	Description
remotehost	Remote hostname (or IP address if DNS hostname is not available, or if DNSLookup is Off).
rfc931	The remote logname of the user.
authuser	The username as which the user has authenticated himself.
[date]	Date and time of the request.
"request"	The request line exactly as it came from the client.
status	The HTTP status code returned to the client.
bytes	The content length of the document transferred.

Squid-Compatible Log Format

The Squid-compatible log format contains one line for each request. The format of each log entry is shown below:

```
timestamp elapsed src-address type/code size method URL
```

Each field is described below:

Field Name	Description
timestamp	The time the request is completed, with millisecond resolution.
elapsed	Elapsed time of the request, in milliseconds.
src-address	IP address of the requesting client.
type	An indication of how the request was handled by the cache. These are described further below.
code	The HTTP reply code when available. For ICP requests this is always "000". If the reply code was not given, it will be logged as "555".
size	For TCP requests, the amount of data written to the client. For UDP requests, the size of the request (in bytes).
method	The request method (GET, POST, etc). For ICP queries, the method is set to ICP_QUERY
URL	The URL of the request. For TCP misses that are handled by ICP, the URL includes the ICP source. The ICP source can be PARENT_HIT, SIBLING_HIT, FIRST_PARENT_MISS, DIRECT (indicating an ICP timeout or no target found), or NONE (indicating ICP was not used for a cache hit). The source is followed by the host name or IP address from which the object was retrieved, and the object MIME type.

Log Entry Types

The type field values are described below:

Value	Description
TCP_	Refers to requests on the HTTP port (3128).
TCP_HIT	A valid copy of the requested object was in the cache.
TCP_MISS	The requested object was not in the cache.
TCP_EXPIRED	The object was in the cache, but it had expired.
TCP_REFRESH	The user forced a refresh ("reload").
TCP_IFMODSINCE	An If-Modified-Since GET request.
TCP_SWAPFAIL	The object was believed to be in the cache, but could not be accessed.
TCP_DENIED	Access was denied for this request.
UDP_	Refers to requests on the ICP port (3130).
UDP_HIT	A valid copy of the requested object was in the cache. This value is also used with ICP queries.
UDP_MISS	The requested object was not in the cache. This value is also used with ICP queries.
UDP_DENIED	Access was denied for this request.

Using a Custom Format

To define your own log format, choose the Custom format string option, and enter the format string using the codes described below:

Format Character	Description
space character	Multiple consecutive spaces are compressed to a single space.
/	A '/
"	A quote character (").
%a	Client IP Address.
%b	Number of bytes returned by the server (or the Cache).

Format Character	Description
%c	The type of object. Usually the MIME-type.
%d	Name or IP address of the server/cache where we retrieved the object.
%e	Number of milliseconds the request took to process.
%g	UNIX type timestamp (GMT).
%h	Client hostname.
%i	The requested URL.
%m	HTTP Method. HTTP Methods are GET, PUT, POST, etc.
%p	Port on the destination server.
%r	First line of the client request.
%s	The code returned by the server (HTTP Code).
%t	Time of the user request (seconds).
%v	Name of the destination server.
%w	What type of action did the CacheOS take to process this request (hit, miss, etc.).
%A	The browser's user agent.
%H	How and where the object was retrieved from the cache hierarchy (DIRECT from the server, PARENT_HIT = from the parent cache, etc.).
%T	Number of seconds the request took to process.
%U	Path component of the requested URL.

Examples for common access log formats are shown below:

Squid log format: %g %e %a %w/%s %b %m %i %u %H/%d %c

NCSA common log format: %h %l %u %t "%r" %s %b

NCSA extended log format: %h %l %u %t "%r" %s %b "%R" "%A"

You can separate the format codes with a space or slash. If you enter multiple spaces, they will be compressed to a single space in the actual access log.

Appendix B

WCCP

Overview

The CacheFlow device can be configured to participate in a WCCP (Web Cache Control Protocol) scheme, where a WCCP-capable router collaborates with a set of WCCP-configured CacheFlow devices to service requests. WCCP is a Cisco-developed protocol. For more information about WCCP, refer to the Cisco website.

The CacheFlow device can support WCCP protocol versions 1 and 2. Only one protocol version can be active on the CacheFlow device at a time. The active WCCP protocol must be matched by the version running on the WCCP router.

WCCP version 1 offers a subset of the functionality offered by version 2. In version 1, a single WCCP router, referred to as the *Home Router*, transparently redirects only TCP port 80 packets (common HTTP traffic) to a maximum of 32 CacheFlow devices. One of the caches participating in the WCCP protocol is automatically elected to configure the Home Router's redirection tables. As such, caches can be transparently added and removed from the WCCP group, without requiring operator intervention as shown in the following figure.

The WCCP version 2 protocol offers the same capabilities as version 1, along with protocol security and multicast protocol broadcasts. In addition, up to 32 WCCP-capable routers can transparently redirect traffic to a set of up to 32 CacheFlow devices. Whereas version 1 was only capable of redirecting TCP port 80 traffic, version 2 WCCP-capable routers can be configured to redirect IP traffic to a set of CacheFlow devices based on various fields within those packets.

This redirection policy and its administrative details comprise a *Service Group*. WCCP version 1 supports only a single Service Group. Version 2 allows routers and caches to participate in multiple simultaneous Service Groups. Thus, routers can transparently redirect IP packets based on their formats. For example, one Service Group could redirect HTTP traffic and another could redirect FTP traffic.

Using WCCP and Advanced Transparent Redirection

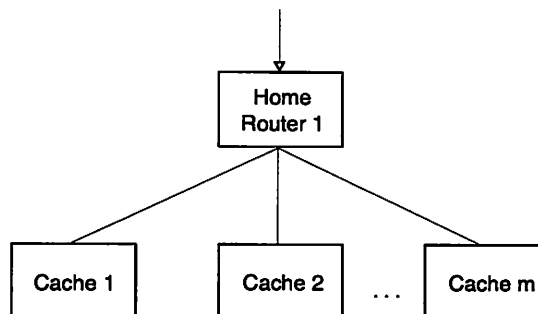
A WCCP-capable router operates in conjunction with CacheFlow devices to transparently redirect traffic to a set of caches which participate in the specified WCCP protocol. IP packets are redirected based on fields within each packet. For instance, WCCP version 1 only redirects destination TCP port 80 (default HTTP traffic) IP packets. The destination IP address is hashed to yield one of 256 buckets within a redirection hash table to determine which cache will be the recipient of the redirected packet. This hash table is configured by a dynamically elected cache participating in the *Service Group*.

In version 2, each service group can be configured to use a security password. Both the routers and caches participating in the *Service Group* use this security password to verify the authenticity of WCCP protocol traffic. Protocol packets that fail the authenticity check are ignored.

Note that it is not recommended that WCCP compliant caches from different vendors participate in the same *Service Group*.

WCCP Version 1

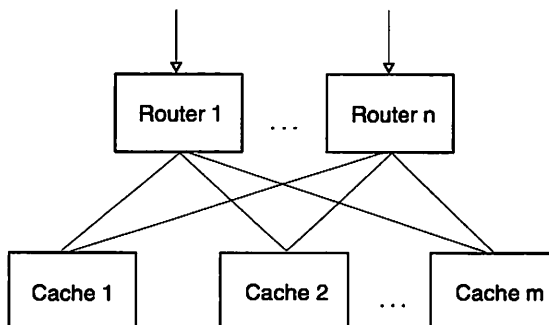
The following figure illustrates a typical WCCP implementation. Each applicable client IP packet received by the *Home Router* is transparently redirected to a cache. A cache from the group is selected to define the Home Router's redirection hash table for all caches. All caches periodically communicate with the Home Router to verify WCCP protocol synchronization and cache availability within the *Service Group*. In return, the Home Router responds to each cache with information as to what caches are available in the Service Group.



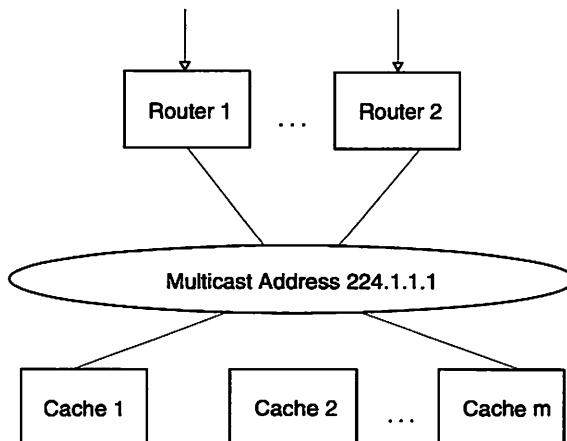
WCCP Version 2

The next figure illustrates a WCCP version 2 implementation using multiple routers and caches. In this scenario, routers 1 through N and

caches 1 through M participate in the same Service Group. As in version 1, a cache from the group is selected to define the redirection hash table in all routers for all caches. All caches periodically communicate with all routers to verify WCCP protocol synchronization and cache and router availability within the Service Group. In return, each router responds to caches with information as to what caches and discovered routers are available in the *Service Group*.



WCCP communication between the routers and the caches can be performed by either directly addressing protocol packets to each router's and cache's IP address (as illustrated in the preceding figure) or by addressing these packets to a common multicast address as illustrated by the following figure:

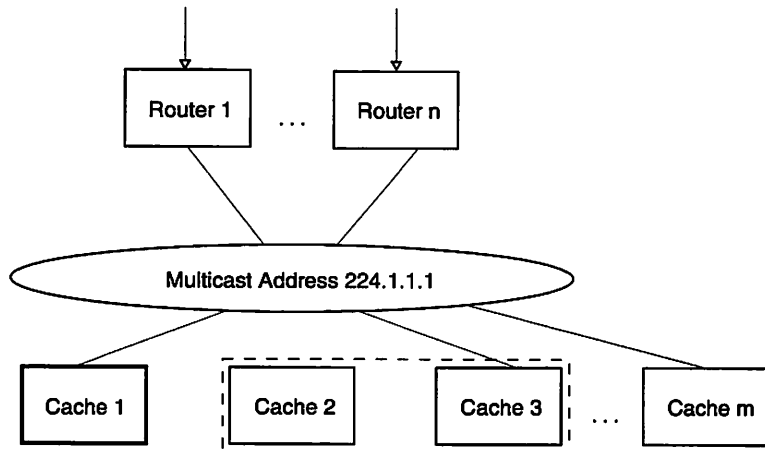


Some of the benefits of using a multicast address include reduced WCCP traffic and the ability to easily add and remove caches and routers from a Service Group without having to reconfigure all Service Group members. Multicast addresses fall within the range 224.0.0.0 to 239.255.255.255.

Multiple Network Cards within a CacheFlow Device

Multiple network cards within a CacheFlow device can participate in the same Service Group. To the routers and other caches, each interface appears as a unique cache. Thus, redirected traffic can be better distributed among network interfaces in a cache.

In the following illustration, Cache 2 and Cache 3 are physically located within the same CacheFlow device using two different network interface cards. Each of these caches will be assigned a unique portion of the redirection hash table and can act as an autonomous unit.



Service Group Security

A password can be applied to a WCCP version 2 *Service Group*. This password must match the configured password within each *Home Router*. Note that is suggested that passwords be exactly 8 characters long.

Distribution of the Redirection Hash Table

As mentioned earlier, a cache within the *Service Group* defines the redirection hash table that it assigns to routers. Each element in this 256 entry table refers to an active cache within the Service Group. When a router receives an IP packet for redirection, it hashes fields within the packet to yield an index within the hash table. Finally, the packet is forwarded to the 'owner' cache for servicing. The proportion of redirection hash table assigned to each cache can be altered to provide a form of load balancing between caches in a *Service Group*.

By default, each cache is assigned roughly an even percentage of the 256 element redirection hash table. Using figure 4 above, all caches would be assigned $1/M$ of the redirection hash table but since **Cache 2** and

Cache 3 are physically located within the same CacheFlow device, the CacheFlow device would actually be assigned $2/M$ of the redirection hash table.

In WCCP version 2, the relative distribution of the redirection hash table can be specified for each cache. Each cache can be assigned a **primary-hash-weight** value (see the “Configuration File Syntax” section below) to determine the proportion of the 256 element hash table to be assigned to a cache. If all caches are configured with a 0 **primary-hash-weight** value (i.e. the default) then each cache will be assigned the same proportion of the redirection hash table. However, if any cache defines a non-zero **primary-hash-weight** then each cache will be assigned a relative proportion of the table. For instance, consider a configuration with five caches whose **primary-hash-weight** is defined as {25, 200, 0, 50, 25}. The total requested weight value is $25+200+0+50+25=300$ and, thus, the proportion of the hash table assigned to each cache will be $25/300$, $200/300$, $0/300$, $50/300$, and $25/300$. Note that since the third cache did not specify a non-zero **primary-hash-weight** it will not be assigned any elements within the redirection hash table and, therefore, will not receive any redirected traffic. Also note that the hash weight can be specified for each caching member within a CacheFlow device. In figure 4 above, **Cache 2** and **Cache 3** could be assigned different weight values.

Alternate Hash Table

In some cases, a web site becomes an Internet ‘hot-spot’ because it receives a disproportionate number of client traffic relative to other sites. This situation can cause a larger request load on a cache relative to its peers within the *Service Group* since the hash element associated with the popular site receives more activity than other hash elements. To balance the redirection traffic load among the caches, a *Service Group* can be configured to use an alternate hash function. A hash element that is identified as a ‘hot-spot’ within the *Service Group* is reconfigured to use an alternate hashing function for computing a new hash table index. The new hashing function can be derived from other components of the IP packet to redirect. Thus, when a router receives an IP packet that hashes to an element flagged as being a ‘hot-spot’, the alternate hash function is computed. The cache as specified by the new index in the redirection hash table will receive the redirected packet.

Each CacheFlow device can dynamically determine a ‘hot-spot’ within its assigned portion of the redirection hash table. Hot-spots are identified as hash elements receiving an excessive amount of traffic over a period of time.

Alternate hash tables are only used for dynamic *Service Groups* that specify **alternate-hash** flags within their **service-flags** (refer to the **service-flags** commands in the “Configuration File Syntax” section below). Note that the the default “**web-cache**” *Service Group* can not use an alternate hash table. Instead, a comparable dynamic *Service Group* must be created.

Configuration File Syntax

The settings configuration file uses three namespaces. The first namespace allows general WCCP commands to be configured. The second and third namespaces allow for specific configuration of *Service Groups*. By default, the Main Namespace is active. Focus can change to the *Service Group* Namespaces by executing a “**service-group** [**web-cache** | **number**]” command. Finally, focus can return to the Main Namespace by executing the “**end**” command. The *Service Group* namespace selected is dependent upon the specified WCCP version. As such, if WCCP version 1 is selected then the version 1 specific *Service Group* Namespace will be used. If no “**wccp version** [**1** | **2**]” is explicitly specified in the configuration file before a *Service Group* is defined then WCCP version 2 is selected.

Main Namespace

wccp [**enable** | **disable**]

This command will enable or disable WCCP. By default, WCCP protocol communication is disabled.

wccp version [**1** | **2**]

This command specifies that the following *Service Group* definitions refers to version 1 or 2 of WCCP. By default, WCCP version 2 *Service Groups* are created unless this command is specified. This command, which can appear at most once within the configuration file, should appear before any *Service Groups* are defined.

service-group [**web-cache** | **number**]

This command introduces the definition for a *Service Group*. The **web-cache** identifier refers to the standard HTTP redirection *Service Group*. If a **number** is specified, then a dynamic *Service Group* is being specified. Once this command is accepted, either the version 1 or version 2 *Service Group* command namespace will become active.

no service-group [**web-cache** | **number**]

This command will destroy a previously defined *Service Group*.

*Version 1 Service Group Namespace***home-router address**

This command allows a home router to be specified for a Service Group. The address must be a dotted decimal value and must not be a multicast address (ie. within the range 224.0.0.0 to 239.255.255.255).

home-router domain-name

This command allows a home router to be specified for a Service Group. A DNS lookup is performed on the domain-name. If the lookup fails then an error will be reported. The domain-name must be a valid domain name string.

interface interfacenumber

This command specifies the network interface number to be associated with the Service Group. Multiple network interfaces within a CacheFlow device may participate within the same Service Group.

no interface interfacenumber

This command removes a network card interface from a Service Group.

end

This command returns focus back to the Main Namespace.

*Version 2 Service Group Namespace***priority number**

This command will set the priority value for the Service Group. The acceptable range is 0 to 255. Note that this command would be used for a dynamic Service Group (ie. one specified as “service-group number”).

protocol number

This command will set the protocol value for the Service Group. The acceptable range is 0 to 255. Note that this command would be used for a dynamic Service Group (ie. one specified as “service-group number”).

service-flags source-ip-hash**service-flags destination-ip-hash****service-flags source-port-hash****service-flags destination-port-hash****service-flags ports-defined****service-flags ports-source****service-flags source-ip-alternate-hash****service-flags destination-ip-alternate-hash**

service-flags source-port-alternate-hash
service-flags destination-port-alternate-hash

These commands set the appropriate bit definitions within the service flags for Service Group.

Note that these commands would be used for a dynamic Service Group (ie. one specified as "service-group number").

no service-flags source-ip-hash
no service-flags destination-ip-hash
no service-flags source-port-hash
no service-flags destination-port-hash
no service-flags ports-defined
no service-flags ports-source
no service-flags source-ip-alternate-hash
no service-flags destination-ip-alternate-hash
no service-flags source-port-alternate-hash
no service-flags destination-port-alternate-hash

These commands reset the appropriate bit definitions within the service flags for Service Group.

Note that these commands would be used for a dynamic Service Group (ie. one specified as "service-group number").

ports number number number number number number number
number number

This command will set the port values for the *Service Group*. Each *number* is a 16 bit decimal value. Note that this command would be used for a dynamic *Service Group* (ie. one specified as "service-group number").

home-router address

This command allows multiple home routers to be specified for a *Service Group*. The *address* must be a dotted decimal value. For WCCP version 2, either a single multicast address (ie. within the range 224.0.0.0 to 239.255.255.255) or up to 32 router IP addresses can be specified.

home-router domain-name

This command allows multiple home routers to be specified for a *Service Group*. A DNS lookup is performed on the *domain-name*. If the lookup fails then an error will be reported. The *domain-name* must be a valid domain name string.

interface interfacenumber

This command specifies the network interface number to be associated with the *Service Group*. Multiple network interfaces within a CacheFlow device may participate within the same *Service Group*.

no interface *interfacenumber*

This command removes a network card interface from a *Service Group*.

password *string*

This command applies a password to a *Service Group*. It is suggested that the password *string* be exactly 8 characters long.

no password

This command removes the password used by a *Service Group*.

primary-hash-weight *interfacenumber value*

This command associates a weight factor of *value* for network interface *interfacenumber* within a *Service Group*. This weighting value is used in version 2 to alter the distribution of the primary hash table.

end

This command returns focus back to the Main Namespace.

Examples

Version 1 Standard HTTP Redirection

Configuring WCCP version 1 on the Router

The following example enables WCCP version 1 on a Cisco router. It is assumed that the router's ethernet interface 0/0 will be used for redirecting traffic to cache members in the *Service Group*.

Router# **configure terminal**

Enter configuration commands, one per line. End with CTRL/Z.

Router(config)# **ip wccp enable**

Router(config)# **interface ethernet 0/0**

Router(config-if)# **ip web-cache redirect**

This configuration simply enables WCCP and assigns redirected traffic to sent out ethernet interface 0/0.

Configuring the CacheFlow Device

To enable the WCCP version 1 *Service Group* within the CacheFlow device, the following configuration file could be loaded.

```

# Enable WCCP to allow WCCP protocol communication between the cache
# and the Home Router.
wccp enable
# A WCCP version 1 Service Group will be configured. Note that the
# following line must be appear before the Service Group definition.
wccp version 1
service-group web-cache
  home-router 90.0.0.90      # specify the address for the router
  interface 0                # network interface 0 will participate
end

```

Version 2 Standard HTTP Redirection

Configuring WCCP version 2 on the Router

The following example will enable the standard HTTP traffic redirection on a WCCP version 2 capable Cisco router.

```

Router# configure terminal
Enter configuration commands, one per line. End with CNTR/Z.
Router(config)# ip wccp web-cache
Router(config)# interface ethernet 0/0
Router(config-if)# ip wccp web-cache redirect out

```

Configuring the CacheFlow Device

To enable the standard WCCP version 2 *Service Group* within the CacheFlow device, the following configuration file could be loaded.

```

# Enable WCCP to allow WCCP protocol communication between the
cache
# and the Home Router.
wccp enable
# By default, the WCCP version 2 protocol is assumed. An explicit
# "wccp version 2" command could be specified here.
service-group web-cache
  home-router 90.0.0.90      # specify the address for the router
  interface 0                # network interface 0 will participate
end

```

Version 2 Standard HTTP Redirection Using a Multicast Address

Configuring WCCP version 2 on the Router

The following example will enable the standard HTTP traffic redirection on a WCCP version 2 capable Cisco router. In this case, WCCP protocol traffic will be directed to the multicast address 224.1.1.1.

```
Router# configure terminal
Enter configuration commands, one per line. End with CNTR/Z.
Router(config)# ip wccp web-cache group-address 224.1.1.1
Router(config)# interface ethernet 0/0
Router(config-if)# ip wccp web-cache group-listen
Router(config-if)# ip wccp web-cache redirect out
```

Configuring the CacheFlow Device

To enable the standard WCCP version 2 *Service Group* within the CacheFlow device, the following configuration file could be loaded. Note that in this example, both network infaces 0 and 1 will participate within the *Service Group*. Both interfaces will send and receive WCCP protocol packets by way of the multicast address.

```
# Enable WCCP to allow WCCP protocol communication between the
cache
# and the Home Router.
wccp enable
# By default, the WCCP version 2 protocol is assumed. An explicit
# "wccp version 2" command could be specified here.
service-group web-cache
  home-router 224.1.1.1 # specify the multicast address
  interface 0          # network interface 0 will participate
  interface 1          # network interface 1 will also participate
end
```

Version 2 Standard HTTP Redirection Using a Security Password

Configuring WCCP version 2 on the Router

The following example will enable standard HTTP traffic redirection on a WCCP version 2 capable Cisco router. A simple eight character password

is configured within the router. This password must match the password configured within the CacheFlow device.

```
Router# configure terminal
```

Enter configuration commands, one per line. End with CNTR/Z.

```
Router(config)# ip wccp web-cache password guesswat
```

```
Router(config)# interface ethernet 0/0
```

```
Router(config-if)# ip wccp web-cache redirect out
```

Configuring the CacheFlow Device

To enable the standard WCCP version 2 *Service Group* within the CacheFlow device, the following configuration file could be loaded.

```
# Enable WCCP to allow WCCP protocol communication between the
cache
```

```
# and the Home Router.
```

```
wccp enable
```

```
# By default, the WCCP version 2 protocol is assumed. An explicit
```

```
# "wccp version 2" command could be specified here.
```

```
service-group web-cache
```

```
home-router 90.0.0.90 # specify the address for the router
```

```
interface 0           # network interface 0 will participate
```

```
password guesswat
```

```
end
```

Version 2 Reverse Proxy Service Group

Configuring WCCP version 2 on the Router

The following example will enable the special reverse proxy *Service Group* on a WCCP version 2 capable Cisco router. This *Service Group* redirects IP packets for TCP destination port 80 traffic by hashing the source IP address.

```
Router# configure terminal
```

Enter configuration commands, one per line. End with CNTR/Z.

```
Router(config)# ip wccp 99
```

```
Router(config)# interface ethernet 0/0
```

```
Router(config-if)# ip wccp 99 redirect out
```

Configuring the CacheFlow Device

To configure the special reverse proxy *Service Group* on the CacheFlow device, a dynamic *Service Group* must be created as illustrated by the following example.

```
# Enable WCCP to allow WCCP protocol communication between the cache
# and the Home Router.
wccp enable
# By default, the WCCP version 2 protocol is assumed. An explicit
# "wccp version 2" command could be specified here.
# Service Group 99 is specially identified within the router as
# representing the reverse proxy service.
service-group 99
  home-router 90.0.0.90 # specify the address for the router
  interface 0          # network interface 0 will participate
  protocol 6           # TCP protocol
  service-flags source-ip-hash # hash based on source IP address
end
```

Version 2 Service Group with Alternate Hashing

Configuring WCCP version 2 on the Router

The following example will enable a special *Service Group* on a WCCP version 2 capable Cisco router that uses alternate hashing when hot-spots are detected. This *Service Group* redirects IP packets by hashing the source IP address.

```
Router# configure terminal
Enter configuration commands, one per line. End with CNTR/Z.
Router(config)# ip wccp 5
Router(config)# interface ethernet 0/0
Router(config-if)# ip wccp 5 redirect out
```

Configuring the CacheFlow Device

To configure this special *Service Group* on the CacheFlow device, a dynamic *Service Group* must be created as illustrated by the following example.

```
# Enable WCCP to allow WCCP protocol communication between the cache
# and the Home Router.
wccp enable
```



```
# By default, the WCCP version 2 protocol is assumed. An explicit
# "wccp version 2" command could be specified here.
# Service Group 5 will be created to redirect standard HTTP traffic
# and use an alternate hash function based on the source IP address
# if necessary.
service-group 5
  home-router 90.0.0.90 # specify the address for router 1
  home-router 90.0.1.5  # specify the address for router 2
  interface 0           # network interface 0 will participate
  protocol 6            # TCP protocol
  # The following two flags specify that a hash function based on the
  # destination IP address should be applied first. If a hot-spot is
  # detected then an alternate hash function using the source IP
  # address should be used.
  service-flags destination-ip-hash
  service-flags source-ip-alternate-hash
end
```

Appendix C

Regular Expressions

Overview

Regular expressions can be used for complex pattern matching. The CacheOS supports regular expressions for URL matching with advanced forwarding and filters. The regular expression support in the CacheOS described in this appendix is based on the Perl-compatible regular expression libraries (PCRE) by Philip Hazel. The text of this appendix is based on the PCRE documentation.

Introduction

A regular expression (or RE) is a pattern that is matched against a subject string from left to right. Most characters stand for themselves in a pattern, and match the corresponding characters in the subject. The power of regular expressions comes from the ability to include alternatives and repetitions in the pattern. These are encoded in the pattern by the use of metacharacters, which do not stand for themselves, but instead are interpreted in some special way. For details of the theory and implementation of regular expressions, consult Jeffrey Friedl's "Mastering Regular Expressions", published by O'Reilly (ISBN 1-56592-257-3).

CacheOS uses a Regular Expression Engine (RE ENGINE) to evaluate regular expressions. In CacheOS, regular expressions can be used for filtering URLs which can match any portion of a URL. More specifically, a URL is considered to be of the form:

```
protocol://domain[:port]/relpath
```

In a CacheOS filter file, a line is considered to be a regular expression if it contains one or more regular expression metacharacters from the following set:

```
\ ^ $ [ | ( ? * + {
```

Portions of the regular expression which match against the protocol and the domain are converted to a canonical form so that matches are performed in a case-insensitive manner. Regular expressions used for filtering URLs can appear anywhere within a filter file; however, the order in which they appear is significant since the first regular expression matched is the one whose associated filter properties are used.

Similarly, in CacheOS Advanced Forwarding, the *icp_host_url_regex* command defines which requests are sent to which cache hosts or cache host groups based on a regular expression match of the requested object's URL.

Regular Expression Syntax

Regular expressions can contain both special and ordinary characters. Most ordinary characters, like 'A', 'a', or '3', are the simplest regular expressions; they simply match themselves. You can concatenate ordinary characters, so 'last' matches the characters 'last'. (In the rest of this section, we'll write RE's in this special font, usually without quotes, and strings to be matched in single quotes.)

Some characters, like | or (, are special. Special characters, called metacharacters, either stand for classes of ordinary characters, or affect how the regular expressions around them are interpreted. The metacharacters are shown below:

Metacharacters	Description
.	(Dot.) In the default mode, this matches any character except a newline
^	(Caret.) Matches the start of the string.
\$	Matches the end of the string.
*	Causes the resulting RE to match 0 or more repetitions of the preceding RE, as many repetitions as are possible. <i>ab*</i> will match 'a', 'ab', or 'a' followed by any number of 'b's.
+	Causes the resulting RE to match 1 or more repetitions of the preceding RE. <i>ab+</i> will match 'a' followed by any non-zero number of 'b's; it will not match just 'a'.
?	Causes the resulting RE to match 0 or 1 repetitions of the preceding RE. <i>ab?</i> will match either 'a' or 'ab'.

Metacharacters	Description
*?, +?, ??	The *, +, and ? qualifiers are all greedy; they match as much text as possible. Sometimes this behavior isn't desired; if the RE <code><.*></code> is matched against <code><H1>title</H1></code> , it will match the entire string, and not just <code><H1></code> . Adding ? after the qualifier makes it perform the match in non-greedy or minimal fashion; as few characters as possible will be matched. Using <code>.?*?</code> in the previous expression will match only <code><H1></code> .
{ <i>m,n</i> }	Causes the resulting RE to match from <i>m</i> to <i>n</i> repetitions of the preceding RE, attempting to match as many repetitions as possible. For example, <code>a{3,5}</code> will match from 3 to 5 'a' characters.
{ <i>m,n</i> }?	Causes the resulting RE to match from <i>m</i> to <i>n</i> repetitions of the preceding RE, attempting to match as <i>few</i> repetitions as possible. This is the non-greedy version of the previous qualifier. For example, on the 6-character string 'aaaaaa', <code>a{3,5}</code> will match 5 'a' characters, while <code>a{3,5}?</code> will only match 3 characters.
\	Either escapes special characters (permitting you to match characters like <code>*?+&\$'</code>), or signals a special sequence; special sequences are discussed below.
[]	Used to indicate a set of characters. Characters can be listed individually, or a range of characters can be indicated by giving two characters and separating them by a '-'. Special characters are not active inside sets. For example, <code>[akm\$]</code> will match any of the characters 'a', 'k', 'm', or '\$'; <code>[a-z]</code> will match any lowercase letter and <code>[a-zA-Z0-9]</code> matches any letter or digit. Character classes such as <code>\w</code> or <code>\S</code> (defined below) are also acceptable inside a range. If you want to include a <code>]</code> or a <code>-</code> inside a set, precede it with a backslash. Characters <i>not</i> within a range can be matched by including a <code>^</code> as the first character of the set; <code>^</code> elsewhere will simply match the <code>^</code> character.

Metacharacters	Description
	A B, where A and B can be arbitrary REs, creates a regular expression that will match either A or B. This can be used inside groups (see below) as well. To match a literal ' ', use \ , or enclose it inside a character class, like [].
(...)	Matches whatever regular expression is inside the parentheses, and indicates the start and end of a group; the contents of a group can be retrieved after a match has been performed, and can be matched later in the string with the <code>\number</code> special sequence, described below. To match the literals '(' or ')', use \(or \), or enclose them inside a character class: [(] [)].

Regular Expression Details

The syntax and semantics of the regular expressions supported by RE ENGINE are described below. Regular expressions are also described in the Perl documentation and in a number of other books, some of which have copious examples. Jeffrey Friedl's "Mastering Regular Expressions", published by O'Reilly (ISBN 1-56592-257-3), covers them in great detail. The description here is intended as reference documentation.

There are two different sets of metacharacters: those that are recognized anywhere in the pattern except within square brackets, and those that are recognized in square brackets. Outside square brackets, the metacharacters are as follows:

- \ general escape character with several uses
- ^ assert start of subject (or line, in multiline mode)
- \$ assert end of subject (or line, in multiline mode)
- .
- [start character class definition
- | start of alternative branch
- (start subpattern
-) end subpattern
- ? extends the meaning of "(also 0 or 1 quantifier also quantifier minimizer

- * 0 or more quantifier
- + 1 or more quantifier
- { start min/max quantifier

Part of a pattern that is in square brackets is called a "character class". In a character class the only metacharacters are:

- \ general escape character
- ^ negate the class, but only if the first character
- indicates character range
-] terminates the character class

The following sections describe the use of each of the metacharacters.

Backslash

The backslash character has several uses. Firstly, if it is followed by a non-alphameric character, it takes away any special meaning that character may have. This use of backslash as an escape character applies both inside and outside character classes.

For example, if you want to match a "*" character, you write "*" in the pattern. This applies whether or not the following character would otherwise be interpreted as a metacharacter, so it is always safe to precede a non-alphameric with "\" to specify that it stands for itself. In particular, if you want to match a backslash, you write "\\".

An escaping backslash can be used to include a whitespace or "#" character as part of the pattern.

A second use of backslash provides a way of encoding non-printing characters in patterns in a visible manner. There is no restriction on the appearance of non-printing characters, apart from the binary zero that terminates a pattern, but when a pattern is being prepared by text editing, it is usually easier to use one of the following escape sequences than the binary character it represents. For example, \a represents "alarm", the BEL character (hex 07).

The handling of a backslash followed by a digit other than 0 is complicated. Outside a character class, RE ENGINE reads it and any following digits as a decimal number. If the number is less than 10, or if there have been at least that many previous capturing left parentheses in the expression, the entire sequence is taken as a *back reference*. A description of how this works is given later, following the discussion of parenthesized subpatterns.

Inside a character class, or if the decimal number is greater than 9 and there have not been that many capturing subpatterns, RE ENGINE re-

reads up to three octal digits following the backslash, and generates a single byte from the least significant 8 bits of the value. Any subsequent digits stand for themselves. For example, `\040` is another way of writing a space

Note that octal values of 100 or greater must not be introduced by a leading zero, because no more than three octal digits are ever read. All the sequences that define a single byte value can be used both inside and outside character classes. In addition, inside a character class, the sequence `"\b"` is interpreted as the backspace character (hex 08). Outside a character class it has a different meaning (see below).

The third use of backslash is for specifying generic character types:

- `\d` any decimal digit
- `\D` any character that is not a decimal digit
- `\s` any whitespace character
- `\S` any character that is not a whitespace character
- `\w` any "word" character
- `\W` any "non-word" character

Each pair of escape sequences partitions the complete set of characters into two disjoint sets. Any given character matches one, and only one, of each pair.

A "word" character is any letter or digit or the underscore character, that is, any character which can be part of a Perl "word". The definition of letters and digits is controlled by RE ENGINE's character tables, and may vary if locale-specific matching is taking place (see "Locale support" above). For example, in the "fr" (French) locale, some character codes greater than 128 are used for accented letters, and these are matched by `\w`.

These character type sequences can appear both inside and outside character classes. They each match one character of the appropriate type. If the current matching point is at the end of the subject string, all of them fail, since there is no character to match.

The fourth use of backslash is for certain simple assertions. An assertion specifies a condition that has to be met at a particular point in a match, without consuming any characters from the subject string. The use of subpatterns for more complicated assertions is described below. The backslashed assertions are

- `\b` word boundary
- `\B` not a word boundary
- `\A` start of subject (independent of multiline mode)

\Z end of subject or newline at end (independent of multiline mode)

\z end of subject (independent of multiline mode)

These assertions may not appear in character classes (but note that "\b" has a different meaning, namely the backspace character, inside a character class).

A word boundary is a position in the subject string where the current character and the previous character do not both match \w or \W (i.e. one matches \w and the other matches \W), or the start or end of the string if the first or last character matches \w, respectively.

The \A, \Z, and \z assertions differ from the traditional circumflex and dollar (described below) in that they only ever match at the very start and end of the subject string, whatever options are set. The difference between \Z and \z is that \Z matches before a newline that is the last character of the string as well as at the end of the string, whereas \z matches only at the end.

Circumflex and Dollar

Outside a character class, in the default matching mode, the circumflex character is an assertion which is true only if the current matching point is at the start of the subject string. Inside a character class, circumflex has an entirely different meaning (see below).

Circumflex need not be the first character of the pattern if a number of alternatives are involved, but it should be the first thing in each alternative in which it appears if the pattern is ever to match that branch. If all possible alternatives start with a circumflex, that is, if the pattern is constrained to match only at the start of the subject, it is said to be an "anchored" pattern. (There are also other constructs that can cause a pattern to be anchored.)

A dollar character is an assertion which is true only if the current matching point is at the end of the subject string, or immediately before a newline character that is the last character in the string (by default).

Dollar need not be the last character of the pattern if a number of alternatives are involved, but it should be the last item in any branch in which it appears. Dollar has no special meaning in a character class.

Full Stop (Period, Dot)

Outside a character class, a dot in the pattern matches any one character in the subject, including a non-printing character, but not (by default) newline. The handling of dot is entirely independent of the handling of

circumflex and dollar, the only relationship being that they both involve newline characters. Dot has no special meaning in a character class.

Square Brackets

An opening square bracket introduces a character class, terminated by a closing square bracket. A closing square bracket on its own is not special. If a closing square bracket is required as a member of the class, it should be the first data character in the class (after an initial circumflex, if present) or escaped with a backslash.

A character class matches a single character in the subject; the character must be in the set of characters defined by the class, unless the first character in the class is a circumflex, in which case the subject character must not be in the set defined by the class. If a circumflex is actually required as a member of the class, ensure it is not the first character, or escape it with a backslash.

For example, the character class `[aeiou]` matches any lower case vowel, while `[^aeiou]` matches any character that is not a lower case vowel. Note that a circumflex is just a convenient notation for specifying the characters which are in the class by enumerating those that are not. It is not an assertion: it still consumes a character from the subject string, and fails if the current pointer is at the end of the string.

When caseless matching is set, any letters in a class represent both their upper case and lower case versions, so for example, a caseless `[aeiou]` matches "A" as well as "a", and a caseless `[^aeiou]` does not match "A", whereas a careful version would.

A class such as `[^a]` will always match a newline.

The minus (hyphen) character can be used to specify a range of characters in a character class. For example, `[d-m]` matches any letter between d and m, inclusive. If a minus character is required in a class, it must be escaped with a backslash or appear in a position where it cannot be interpreted as indicating a range, typically as the first or last character in the class. It is not possible to have the character "]" as the end character of a range, since a sequence such as `[w-]` is interpreted as a class of two characters. The octal or hexadecimal representation of "]" can, however, be used to end a range.

Ranges operate in ASCII collating sequence. They can also be used for characters specified numerically, for example `[\000-\037]`. If a range that includes letters is used when caseless matching is set, it matches the letters in either case. For example, `[W-c]` is equivalent to `[^\^_ 'wxyzabc]`, matched caselessly, and if character tables for the "fr" locale are in use, `[\xc8-\xcb]` matches accented E characters in both cases.

The character types `\d`, `\D`, `\s`, `\S`, `\w`, and `\W` may also appear in a character class, and add the characters that they match to the class. For example, `[dABCDEF]` matches any hexadecimal digit. A circumflex can conveniently be used with the upper case character types to specify a more restricted set of characters than the matching lower case type. For example, the class `^[W_]` matches any letter or digit, but not underscore.

All non-alphanumeric characters other than `\`, `-`, `^` (at the start) and the terminating `]` are non-special in character classes, but it does no harm if they are escaped.

Vertical Bar

Vertical bar characters are used to separate alternative patterns. For example, the pattern

```
gilbert|sullivan
```

matches either "gilbert" or "sullivan". Any number of alternatives may appear, and an empty alternative is permitted (matching the empty string). The matching process tries each alternative in turn, from left to right, and the first one that succeeds is used. If the alternatives are within a subpattern (defined below), "succeeds" means matching the rest of the main pattern as well as the alternative in the subpattern.

Subpatterns

Subpatterns are delimited by parentheses (round brackets), which can be nested. Marking part of a pattern as a subpattern does two things:

1. It localizes a set of alternatives. For example, the pattern

```
cat(aract|erpillar|)
```

matches one of the words "cat", "cataract", or "caterpillar". Without the parentheses, it would match "cataract", "erpillar" or the empty string.

2. It sets up the subpattern as a capturing subpattern (as defined above). When the whole pattern matches, that portion of the subject string that matched the subpattern is passed back to the caller via the *ovector* argument of *RE Engine_exec()*. Opening parentheses are counted from left to right (starting from 1) to obtain the numbers of the capturing subpatterns.

For example, if the string "the red king" is matched against the pattern

```
the ((red|white) (king|queen))
```

the captured substrings are "red king", "red", and "king", and are numbered 1, 2, and 3.

The fact that plain parentheses fulfil two functions is not always helpful. There are often times when a grouping subpattern is required without a capturing requirement. If an opening parenthesis is followed by "?:", the subpattern does not do any capturing, and is not counted when computing the number of any subsequent capturing subpatterns. For example, if the string "the white queen" is matched against the pattern

```
the ((?:red|white) (king|queen))
```

the captured substrings are "white queen" and "queen", and are numbered 1 and 2. The maximum number of captured substrings is 99, and the maximum number of all subpatterns, both capturing and non-capturing, is 200.

As a convenient shorthand, if any option settings are required at the start of a non-capturing subpattern, the option letters may appear between the "?" and the ":". Thus the two patterns

```
(?:saturday|sunday)
(?:i)saturday|sunday
```

match exactly the same set of strings. Because alternative branches are tried from left to right, and options are not reset until the end of the subpattern is reached, an option setting in one branch does affect subsequent branches, so the above patterns match "SUNDAY" as well as "Saturday".

Repetition

Repetition is specified by quantifiers, which can follow any of the following items:

- a single character, possibly escaped the . metacharacter
- a character class
- a back reference (see next section)
- a parenthesized subpattern (unless it is an assertion - see below)

The general repetition quantifier specifies a minimum and maximum number of permitted matches, by giving the two numbers in curly brackets (braces), separated by a comma. The numbers must be less than 65536, and the first must be less than or equal to the second. For example:

`z{2,4}`

matches "zz", "zzz", or "zzzz". A closing brace on its own is not a special character. If the second number is omitted, but the comma is present, there is no upper limit; if the second number and the comma are both omitted, the quantifier specifies an exact number of required matches. Thus

`[aeiou]{3,}`

matches at least 3 successive vowels, but may match many more, while

`\d{8}`

matches exactly 8 digits. An opening curly bracket that appears in a position where a quantifier is not allowed, or one that does not match the syntax of a quantifier, is taken as a literal character. For example, `{,6}` is not a quantifier, but a literal string of four characters.

The quantifier `{0}` is permitted, causing the expression to behave as if the previous item and the quantifier were not present. For convenience (and historical compatibility) the three most common quantifiers have single-character abbreviations:

- `*` is equivalent to `{0,}`
- `+` is equivalent to `{1,}`
- `?` is equivalent to `{0,1}`

It is possible to construct infinite loops by following a subpattern that can match no characters with a quantifier that has no upper limit, for example:

`(a?)*`

Earlier versions of Perl gave an error at compile time for such patterns. However, because there are cases where this can be useful, such patterns are now accepted, but if any repetition of the subpattern does in fact match no characters, the loop is forcibly broken.

By default, the quantifiers are "greedy", that is, they match as much as possible (up to the maximum number of permitted times), without causing the rest of the pattern to fail. The classic example of where this gives problems is in trying to match comments in C programs. These appear between the sequences `/*` and `*/` and within the sequence, individual `*` and `/` characters may appear. An attempt to match C comments by applying the pattern

```
/^.*\*/
```

to the string

```
/^ first command */ not comment /* second comment */
```

fails, because it matches the entire string due to the greediness of the `.*` item.

However, if a quantifier is followed by a question mark, then it ceases to be greedy, and instead matches the minimum number of times possible, so the pattern

```
/^.*?\*/
```

does the right thing with the C comments. The meaning of the various quantifiers is not otherwise changed, just the preferred number of matches. Do not confuse this use of question mark with its use as a quantifier in its own right. Because it has two uses, it can sometimes appear doubled, as in

```
\d??\d
```

which matches one digit by preference, but can match two if that is the only way the rest of the pattern matches.

When a parenthesized subpattern is quantified with a minimum repeat count that is greater than 1 or with a limited maximum, more store is required for the compiled pattern, in proportion to the size of the minimum or maximum.

If a pattern starts with `.*` then it is implicitly anchored, since whatever follows will be tried against every character position in the subject string. RE ENGINE treats this as though it were preceded by `\A`.

When a capturing subpattern is repeated, the value captured is the substring that matched the final iteration. For example, after

```
(tweedle[dume]{3}\s*)+
```

has matched "tweedledum tweedledee" the value of the captured substring is "tweedledee". However, if there are nested capturing subpatterns, the corresponding captured values may have been set in previous iterations. For example, after

```
/a(b)+/
```

matches "aba" the value of the second captured substring is "b".

Back References

Outside a character class, a backslash followed by a digit greater than 0 (and possibly further digits) is a back reference to a capturing subpattern earlier (i.e. to its left) in the pattern, provided there have been that many previous capturing left parentheses.

However, if the decimal number following the backslash is less than 10, it is always taken as a back reference, and causes an error only if there are not that many capturing left parentheses in the entire pattern. In other words, the parentheses that are referenced need not be to the left of the reference for numbers less than 10. See the section entitled "Backslash" above for further details of the handling of digits following a backslash.

A back reference matches whatever actually matched the capturing subpattern in the current subject string, rather than anything matching the subpattern itself. So the pattern

```
(sens)respons)e and \1bility
```

matches "sense and sensibility" and "response and responsibility", but not "sense and responsibility". If careful matching is in force at the time of the back reference, then the case of letters is relevant. For example,

```
((?i)rah)\s+1
```

matches "rah rah" and "RAH RAH", but not "RAH rah", even though the original capturing subpattern is matched caselessly.

There may be more than one back reference to the same subpattern. If a subpattern has not actually been used in a particular match, then any back references to it always fail. For example, the pattern

```
(a(bc))\2
```

always fails if it starts to match "a" rather than "bc". Because there may be up to 99 back references, all digits following the backslash are taken as part of a potential back reference number. If the pattern continues with a digit character, then some delimiter must be used to terminate the back reference.

A back reference that occurs inside the parentheses to which it refers fails when the subpattern is first used, so, for example, `(a\1)` never matches. However, such references can be useful inside repeated subpatterns. For example, the pattern

```
(alb\1)+
```

matches any number of "a"s and also "aba", "ababaa" etc. At each iteration of the subpattern, the back reference matches the character string corresponding to the previous iteration. In order for this to work, the pattern must be such that the first iteration does not need to match the back reference. This can be done using alternation, as in the example above, or by a quantifier with a minimum of zero.

Assertions

An assertion is a test on the characters following or preceding the current matching point that does not actually consume any characters. The simple assertions coded as `\b`, `\B`, `\A`, `\Z`, `\z`, `^` and `$` are described above. More complicated assertions are coded as subpatterns. There are two kinds: those that look ahead of the current position in the subject string, and those that look behind it.

An assertion subpattern is matched in the normal way, except that it does not cause the current matching position to be changed. Lookahead assertions start with `(?=` for positive assertions and `(?!` for negative assertions. For example,

```
\w+(?=;)
```

matches a word followed by a semicolon, but does not include the semicolon in the match, and

```
foo(?!bar)
```

matches any occurrence of "foo" that is not followed by "bar". Note that the apparently similar pattern

```
(?!foo)bar
```

does not find an occurrence of "bar" that is preceded by something other than "foo"; it finds any occurrence of "bar" whatsoever, because the assertion `(?!foo)` is always true when the next three characters are "bar". A lookbehind assertion is needed to achieve this effect.

Lookbehind assertions start with `(?<=` for positive assertions and `(?<!` for negative assertions. For example,

```
(?<!foo)bar
```

does find an occurrence of "bar" that is not preceded by "foo". The contents of a lookbehind assertion are restricted such that all the strings it matches must have a fixed length. However, if there are several alternatives, they do not all have to have the same fixed length. Thus

```
(?<=bullock|donkey)
```

is permitted, but

```
(?<!dogs?!cats?)
```

causes an error at compile time. Branches that match different length strings are permitted only at the top level of a lookbehind assertion. This is an extension compared with Perl 5.005, which requires all branches to match the same length of string. An assertion such as

```
(?<=ab(c|de))
```

is not permitted, because its single branch can match two different lengths, but it is acceptable if rewritten to use two branches:

```
(?<=abclabde)
```

The implementation of lookbehind assertions is, for each alternative, to temporarily move the current position back by the fixed width and then try to match. If there are insufficient characters before the current position, the match is deemed to fail.

Assertions can be nested in any combination. For example,

```
(?<=(?<!foo)bar)baz
```

matches an occurrence of "baz" that is preceded by "bar" which in turn is not preceded by "foo".

Assertion subpatterns are not capturing subpatterns, and may not be repeated, because it makes no sense to assert the same thing several times. If an assertion contains capturing subpatterns within it, these are always counted for the purposes of numbering the capturing subpatterns in the whole pattern. Substring capturing is carried out for positive assertions, but it does not make sense for negative assertions.

Assertions count towards the maximum of 200 parenthesized subpatterns.

Once-Only Subpatterns

With both maximizing and minimizing repetition, failure of what follows normally causes the repeated item to be re-evaluated to see if a different number of repeats allows the rest of the pattern to match. Sometimes it is useful to prevent this, either to change the nature of the match, or to cause

it fail earlier than it otherwise might, when the author of the pattern knows there is no point in carrying on.

Consider, for example, the pattern `\d+foo` when applied to the subject line

`123456bar`

After matching all 6 digits and then failing to match "foo", the normal action of the matcher is to try again with only 5 digits matching the `\d+` item, and then with 4, and so on, before ultimately failing. Once-only subpatterns provide the means for specifying that once a portion of the pattern has matched, it is not to be re-evaluated in this way, so the matcher would give up immediately on failing to match "foo" the first time. The notation is another kind of special parenthesis, starting with `(?>` as in this example:

`(?>\d+)bar`

This kind of parenthesis "locks up" the part of the pattern it contains once it has matched, and a failure further into the pattern is prevented from backtracking into it. Backtracking past it to previous items, however, works as normal.

An alternative description is that a subpattern of this type matches the string of characters that an identical standalone pattern would match, if anchored at the current point in the subject string.

Once-only subpatterns are not capturing subpatterns. Simple cases such as the above example can be thought of as a maximizing repeat that must swallow everything it can. So, while both `\d+` and `\d+?` are prepared to adjust the number of digits they match in order to make the rest of the pattern match, `(?>\d+)` can only match an entire sequence of digits.

This construction can of course contain arbitrarily complicated subpatterns, and it can be nested.

Conditional Subpatterns

It is possible to cause the matching process to obey a subpattern conditionally or to choose between two alternative subpatterns, depending on the result of an assertion, or whether a previous capturing subpattern matched or not. The two possible forms of conditional subpattern are

`(?(condition)yes-pattern)`
`(?(condition)yes-pattern|no-pattern)`

If the condition is satisfied, the yes-pattern is used; otherwise the no-pattern (if present) is used. If there are more than two alternatives in the subpattern, a compile-time error occurs.

There are two kinds of condition. If the text between the parentheses consists of a sequence of digits, then the condition is satisfied if the capturing subpattern of that number has previously matched. Consider the following pattern, which contains non-significant white space to make it more readable and to divide it into three parts for ease of discussion:

```
( \ )?  [^)]+  (? (1) \ )
```

The first part matches an optional opening parenthesis, and if that character is present, sets it as the first captured substring. The second part matches one or more characters that are not parentheses. The third part is a conditional subpattern that tests whether the first set of parentheses matched or not. If they did, that is, if subject started with an opening parenthesis, the condition is true, and so the yes-pattern is executed and a closing parenthesis is required. Otherwise, since no-pattern is not present, the subpattern matches nothing. In other words, this pattern matches a sequence of non-parentheses, optionally enclosed in parentheses.

If the condition is not a sequence of digits, it must be an assertion. This may be a positive or negative lookahead or lookbehind assertion. Consider this pattern, again containing non-significant white space, and with the two alternatives on the second line:

```
(?(?=[^a-z]*[a-z])
 \d{2}[a-z]{3}-\d{2} | \d{2}-\d{2}-\d{2} )
```

The condition is a positive lookahead assertion that matches an optional sequence of non-letters followed by a letter. In other words, it tests for the presence of at least one letter in the subject. If a letter is found, the subject is matched against the first alternative; otherwise it is matched against the second. This pattern matches strings in one of the two forms dd-aaa-dd or dd-dd-dd, where aaa are letters and dd are digits.

Comments

The sequence `(?#` marks the start of a comment which continues up to the next closing parenthesis. Nested parentheses are not permitted. The characters that make up a comment play no part in the pattern matching at all.

Performance

Certain items that may appear in patterns are more efficient than others. It is more efficient to use a character class like `[aeiou]` than a set of alternatives such as `(a|e|i|o|u)`. In general, the simplest construction that provides the required behavior is usually the most efficient.

RE ENGINE Differences From PERL

The differences described here are with respect to Perl 5.005.

1. Normally "space" matches space, formfeed, newline, carriage return, horizontal tab, and vertical tab. Perl 5 no longer includes vertical tab in its set of whitespace characters. The `\v` escape that was in the Perl documentation for a long time was never in fact recognized. However, the character itself was treated as whitespace at least up to 5.002. In 5.004 and 5.005 it does not match `\s`.
2. RE ENGINE does not allow repeat quantifiers on lookahead assertions. Perl permits them, but they do not mean what you might think. For example, `(?!a){3}` does not assert that the next three characters are not "a". It just asserts that the next character is not "a" three times.
3. Capturing subpatterns that occur inside negative lookahead assertions are counted, but their entries in the offsets vector are never set. Perl sets its numerical variables from any such patterns that are matched before the assertion fails to match something (thereby succeeding), but only if the negative lookahead assertion contains just one branch.
4. Though binary zero characters are supported in the subject string, they are not allowed in a pattern string because it is passed as a normal C string, terminated by zero. The escape sequence `"\0"` can be used in the pattern to represent a binary zero.
5. The following Perl escape sequences are not supported: `\l`, `\u`, `\L`, `\U`, `\E`, `\Q`. In fact these are implemented by Perl's general string-handling and are not part of its pattern matching engine.
6. The Perl `\G` assertion is not supported as it is not relevant to single pattern matches.
7. RE ENGINE does not support the `(?{code})` construction.

8. There are at the time of writing some oddities in Perl 5.005_02 concerned with the settings of captured strings when part of a pattern is repeated. For example, matching "aba" against the pattern `/^ (a (b) ?) + $/` sets \$2 to the value "b", but matching "aabbaa" against `/^ (aa (bb) ?) + $/` leaves \$2 unset. However, if the pattern is changed to `/^ (aa (b (b) ) ?) + $/` then \$2 (and \$3) get set. In Perl 5.004 \$2 is set in both cases, and that is also true of RE ENGINE.
9. Another as yet unresolved discrepancy is that in Perl 5.005_02 the pattern `/^ (a) ? ( ? (1) a | b) + $/` matches the string "a", whereas in RE ENGINE it does not. However, in both Perl and RE ENGINE `/^ (a) ? a/` matched against "a" leaves \$1 unset.
10. RE ENGINE provides some extensions to the Perl regular expression facilities: Although lookbehind assertions must match fixed length strings, each alternative branch of a lookbehind assertion can match a different length of string. Perl 5.005 requires them all to have the same length.

Regular Expression Examples

Common examples of URL matching used with the CacheOS are listed below:

`.*://.*\.edu.*`

Matches all URLs in the EDU domain.

`http://.*\.(msnbc|cnn|espn)\.com.*`

Matches URLs containing either msnbc.com, cnn.com, or espn.com.

`ftp://.*`

Matches all FTP requests.

`.*cacheflow\.com:(8081|8084).*`

Matches all requests to port 8081 or port 8084 in the cacheflow.com domain.

Index

?

?, 35

A

access log, 122
 log format, 125
 upload schedule, 124
 upload site, 123
access restrictions, 119
access-log
 configuring, 56
 format, 56
 upload, 57
 uploading, 107
ACL, 119
acquire-utc, 95
Active Caching, 1
active clients, 164
advertising objects, 67, 146
auto-configuration, 7

B

bandwidth utilization, 136, 138
baud rate, 34
bypass list, 153
 central, 59, 154
 gateway, 153
 local, 58, 153
bypass-list
 configure, 58
 loading, 99
bytes served, 163

C

cache
 clearing, 96, 129
 restarting, 103
cache contents, 167
cache efficiency, 166
cache_host, 26
cache_host_domain, 22, 23, 24, 25, 188
cache_host_ip, 29
CacheFlow Web cache
 name, 73, 115
CacheOS, 1
caching
 advertising objects, 67, 146
 configuring, 61
 filtering, 67, 146
 FTP, 138
 maximum object size, 138
 multiple servers, 71
 negative responses, 138
 network bandwidth
 utilization, 136, 138
 refresh policies, 137
 showing, 41
 transparent, 3
caching options, 138
clear-cache, 96
client
 configuring, 5
 configuring Internet Explorer, 13, 14, 15
 configuring Netscape, 10, 11
 custom PAC file, 7
 default PAC file, 7

- notifying, 9
- PAC configuration, 7
- proxy configuration, 7
- using transparency, 3
- client instructions, 6, 116
- command line interface, 33
 - enable mode, 36
 - help, 35, 36
 - logging in, 35
 - modes, 36
 - password, 35
 - standard mode, 36
 - Telnet, 33
 - username, 35
- community strings, 87, 121
- configuration
 - showing, 38, 42
- configure
 - access-log, 56
 - bypass-list, 58
 - caching, 61
 - direct-deny-list, 63
 - dns, 64
 - filter-list, 67
 - forwarding, 71
 - ftp caching, 72
 - hostname, 73
 - http-proxy-port, 74
 - icmp, 75
 - interface, 76
 - ip-default-gateway, 78
 - line-vty, 79
 - management-port, 80
 - no, 81
 - ntp, 82
 - restart, 83
 - rip, 85
 - security, 86
 - snmp, 87
 - socks-machine-id, 89
 - static-routes, 90
 - upgrade-path, 91
 - web-management, 93
- configure command, 54
- configuring
 - event-log, 65
- configuring WCCP support, 92, 141
- console port

- showing, 50
- content distribution
 - showing, 43
- contents, 167
- custom PAC file, 7

D

- data bits, 34
- default PAC file, 7
- defaults
 - restoring, 104
- direct or deny list, 63, 145
- direct-deny-list
 - configuring, 63
 - loading, 99
- disk
 - taking offline, 100
- disk caching
 - Internet Explorer, 13, 15, 16
 - Netscape, 11, 12
- disk resources, 165
- display, 97, 98
- dns
 - configuring, 64
 - showing, 46
- DNS cache, 2, 4
 - purging, 101, 129
- DNS server
 - name imputing, 133
 - setting, 113
- downloading
 - system upgrade, 130
- download-paths
 - showing, 45

E

- efficiency, 166
 - showing, 47
- email
 - client instructions, 9
 - enable mode, 36, 37
 - event log, 131
 - event notification, 126
 - event-log
 - configuring, 65

F

- failures, 30
- filter list, 70, 149
 - central, 69, 149
 - local, 69, 148
- filtering, 4, 67, 146
- filter-list
 - configuring, 67
 - loading, 99
- FindProxyForURL, 8
 - example, 8
- flow control, 34
- forwarding, 17, 71
 - advanced, 18, 20
 - configuring, 71
 - failures, 30
 - groups, 21
 - order, 31
 - simple, 17
- FTP, 4
 - caching, 138
- ftp caching
 - configuring, 72

G

- gateway
 - showing, 49

H

- hierarchical cache, 17
- hostname
 - configuring, 73
- HTTP requests
 - IP port, 114
- http-proxy-port
 - configuring, 74

I

- icp
 - configuring, 75
- ICP, 19
 - configuring, 25
 - failures, 30

- restricting access, 27
- icp-settings
 - loading, 99
- interface
 - configuring, 76
 - showing, 48
- Internet Explorer
 - configuring, 13, 14, 15
- IP address
 - setting, 112
- IP ports, 114
- ip-default-gateway
 - configuring, 78
 - showing, 49

L

- last modified date, 138
- line-vty
 - configuring, 79
- link settings, 112
- load
 - list, 99
- log format, 125, 169
- logging
 - access logging, 122
 - event log, 131
 - log format, 125
 - upload schedule, 124
 - upload site, 123

M

- maximum object size, 138
- management
 - advanced tasks, 111
 - basic tasks, 110
 - SNMP, 87, 121
 - statistics, 111
- management commands, 94
- Management Console
 - IP port, 114
- Management Console password
 - setting, 118
- management-port
 - configuring, 80
- memory resources, 165

N

- name, 73, 115
- name imputing, 133
- negative responses, 138
- Netscape
 - configuring, 10, 11
- network adapter
 - configuring, 112
 - link settings, 112
- network bandwidth utilization, 136, 138
- no
 - configure command, 81
- notifying clients, 9
- ntp
 - configuring, 82

O

- object prefetching, 2
- objects served, 163
- offline-disk, 100

P

- PAC configuration, 7
- PAC file
 - creating, 7
 - custom, 7
 - default, 7
- parity, 34
- ping, 102
- ports
 - showing, 50
- prefetching, 2
- proxy configuration, 7
- Proxy User Authentication, 86, 120
- purge-dns-cache, 101
- purging the DNS cache, 129

R

- refresh date, 138
- refresh policies, 137
- regular expressions, 187

- resources, 165
 - showing, 51
- restart, 103
 - configuring, 83
- restarting, 127
- restore-defaults, 104
- restricting access, 119
- return to sender, 84
- rip
 - configuring, 85
- RIP, 156
 - configuration, 156
- router, 3
- routing
 - direct or deny, 63, 145
 - static routes, 90, 151
 - using a SOCKS server, 89, 143

S

- security
 - configuring, 86
- Serial Console
 - command line interface, 34
 - connecting a terminal, 34
 - setup console, 34
- serial port, 34
- setup console, 34
- show caching, 41
- show command, 38
- show configuration, 42
- show content distribution, 43
- show dns, 46
- show download-paths, 45
- show efficiency, 47
- show interface, 48
- show ip-default-gateway, 49
- show parameters, 38
- show ports, 50
- show resources, 51
- show snmp, 52
- show status, 53
- snmp
 - configuring, 87
- SNMP
 - enabling, 87, 121
- snmp configuration

- showing, 52
- SOCKS, 89, 142, 143
- socks-machine-id
 - configuring, 89
- Squid-compatible log format, 170
- standard mode, 36, 37
- static routes, 151
- static-routes
 - configuring, 90
- static-route-table
 - loading, 99
- statistics, 111
 - active clients, 164
 - bytes served, 163
 - cache contents, 167
 - cache efficiency, 166
 - objects served, 163
 - resources, 165
 - system status, 162
- status, 162
 - showing, 39, 53
- stop bits, 34
- switch, 3
- syslog, 65, 132
- system cache
 - clearing, 129
- system status, 162
- system upgrade, 130

T

- Telnet, 33

- terminal, 34
 - settings, 34
- terminal emulator, 34
- terminal server, 34
- test, 105
- time
 - setting, 82, 117
- traceroute, 106
- transparent caching, 3

U

- upgrade
 - loading, 99
- upgrade-path
 - configuring, 91
- upgrading, 130
- upload schedule, 124
- upload site, 123
- upload-access-log, 107
- URL
 - displaying, 97, 98
 - regular expressions, 187
- using syslog monitoring, 65, 132
- UTC, 82, 117

W

- WCCP support, 92, 141
- web-management
 - configuring, 93